

# Inhaltsverzeichnis

<b>Verfügbare Plugins</b> .....	1
Device Plugins .....	1
Exchange Plugins .....	1
Interface Plugins .....	1
<b>Aktivierung</b> .....	2
<b>Zustandsautomat</b> .....	2
<b>Fehlerdiagnose</b> .....	2
<b>Speicherort</b> .....	3
<b>Klassifizierung</b> .....	3
Device Plugins .....	3
Exchange Plugins .....	3
Interface Plugins .....	4
<b>Konfiguration</b> .....	4
Verwenden einer Anwendung .....	4
Verwenden einer Konfigurationsdatei .....	4
Speicherort .....	5
Struktur .....	5
Objektattribute .....	5
Synchronisation .....	6



# Plugins

Für CoDaBix® sind verschiedene Plugins verfügbar, die durch ihre Klassifizierung gruppiert sind. Je nach Kategorie bietet das Plugin eine spezielle Reihe von Diensten, Entitäten und Nodes an. Einige Plugins können Konfigurationsdateien zur Verfügung stellen und zusätzlich, je nachdem wie es die Umstände erfordern, eine Konfigurationsanwendung.

## Verfügbare Plugins

### Device Plugins

- Allgemein
- AKLAN Device Plugin
- Allen-Bradley Device Plugin
- EUROMAP Device Plugin
- H1 Device Plugin
- Melsec QJ Device Plugin
- Modbus Device Plugin
- Omron Device Plugin
- OPC UA Client Device Plugin
- S7 Device Plugin
- MQTT Client Device Plugin
- UniPi Device Plugin

### Exchange Plugins

- Allgemein
- CSV Exchange Plugin
- Database Plugin
- SQL Exchange Plugin
- XML Exchange Plugin

### Interface Plugins

- Allgemein
- OPC UA Server Interface Plugin
- REST Interface Plugin
- Script Interface Plugin

## Aktivierung

Während des Startvorgangs von CoDaBix® werden alle in dem Pluginverzeichnis befindlichen Plugins (<CodabixPluginsDir>) rekursiv durchlaufen, geladen, instanziiert und in alphabetischer Reihenfolge gestartet. Jedes Plugin, das sich nicht in dem Pluginverzeichnis oder in einem Unterverzeichnis befindet, wird weder geladen noch vom CoDaBix® Plugin Manager gestartet.

Während die CoDaBix® Engine gestartet wird, ist es möglich, ein Plugin neu zu starten. Dies geschieht, indem eine Anfrage mittels Plugin-Proxyinstanz an den CoDaBix® Plugin Manager gesendet wird (beachten Sie, dass diese Funktion ohne Vorankündigung die Änderungen vornimmt).

## Zustandsautomat

Während des Startvorgangs und während der Laufzeit von CoDaBix® kann das Plugin durch verschiedene Zustände laufen. Jeder Status beschränkt den möglichen Übergang von einem Zustand in den anderen, während einige von ihnen aufeinanderfolgend auftreten. Die Plugininstanz bleibt selbst in dem Zustand Created, nachdem der CoDaBix® Plugin-Manager diese geladen und instanziiert hat.

Nachdem der Plugin Manager alle Plugins geladen und instanziiert hat, ruft die Startsequenz die Startsequenz der Plugins auf. Während des Startvorgangs wird der Status auf Starting gesetzt. Nach erfolgreichem Start des Plugins wird der Status auf Started gewechselt. Wenn das Plugin nicht gestartet werden kann, wird der Status wieder auf den Wert, in dem er vor Starting war, gesetzt. Nur Plugins mit dem Status Started können gestoppt werden.

Ein Plugin wird während der Neustartsequenz oder beim Herunterfahren des CoDaBix® Plugin Managers gestoppt. Das Plugin verarbeitet dann seine Stopsequenz (nur im Zustand Started). Während dieser Sequenz ändert sich der Pluginstatus zu Stopping. Bei erfolgreichem Beenden des Plugins wird der Status auf Stopped gesetzt. Wenn das Plugin nicht gestoppt werden kann, fällt es in seinen Zustand zurück, bevor es zu Stopping geändert wurde.

## Fehlerdiagnose

Die zur Verfügung gestellten Diagnoseinformationen von einem Plugin hängen von der Klassifizierung des Plugins ab und ob das Plugin spezifische Diagnoseinformationen selbst zur Verfügung stellt. Das [S7 Device Plugin](#) z.B. ist ein [Device Plugin](#) und stellt Statusinformationen und andere Diagnoseinformationen über sein CoDaBix® Device, seinen CoDaBix® Device Channel und seine S7 Device Variablen bereit.

## Speicherort

Alle Plugin-spezifischen Dateien werden in einem eigenen Pluginverzeichnis innerhalb des Plugin-Verzeichnisses abgelegt, welches sich im CoDaBix® Installationsverzeichnis befindet (ausgewählt während der Installation). Die folgende Liste zeigt die Hierarchie:

- <CodabixInstallDir>
  - plugins\
    - <PluginName>
      - <PluginAssembly>
      - ...

Zum Beispiel:

- C:\Program Files\Traeger\CoDaBix\
  - plugins\
    - S7Device\
      - CoDaBix.S7DevicePlugin.dll
      - ...

## Klassifizierung

### Device Plugins

Alle als Device Plugin klassifizierten Plugins binden physische oder logische Geräte in CoDaBix® ein. Ein Gerät selbst kann jede Art von Ressource sein, die durch das Kanalmodell, welches durch das CoDaBix® [Device Modell](#) definiert ist, für CoDaBix® zugänglich gemacht wird.

Im Allgemeinen definiert ein Device Plugin einen spezialisierten Typen eines CoDaBix® Devices unter Verwendung des CoDaBix® Device Modells (für mehr Informationen siehe [Device Plugins](#)) und stellt dadurch die nötige Zugangsschicht für diese Art von Gerät bereit.

### Exchange Plugins

Alle als Exchange Plugin klassifizierten Plugins, verbinden Storage Engines mit CoDaBix®. Eine Storage Engine selbst kann jede Art von Datenbank-Managementsystem sein (DBMS), dessen Instanzen durch das relationale Modell unter Verwendung vom CoDaBix® [Exchange Modell](#) zugänglich sind. Die Storage Engine selbst muss mindestens ein primitives relationales Modell zur Verfügung stellen, z.B. eine Tabelle (= Einheit in DBMS).

Im Allgemeinen definiert ein Exchange Plugin einen spezialisierten Typen eines CoDaBix® Stages unter Verwendung des CodaBix® Storage Models (weitere Informationen finden Sie unter [Exchange Plugins](#)) und stellt dadurch die notwendige Zugangsschicht für die Art von Storage Engine bereit.

### Interface Plugins

Alle als Interface Plugin klassifizierten Plugins verbinden CoDaBix® mit anderen Plattformen und Technologien. Das Interface kann jede Art von Sprache, Dienst, Protokoll etc. sein. Das API Modell muss nur die spezielle Umgebung des Interfaces auf die von CoDaBix® definierte API-Schnittstelle abbilden.

Im Allgemeinen definiert ein Interface Plugin einen spezialisierten Typen einer Plattform oder Technologie-API (weitere Informationen unter [Interface Plugins](#)) und stellt dadurch die notwendige Zugangsschicht für die CoDaBix® Programmierschnittstelle bereit.

## Konfiguration

### Verwenden einer Anwendung

#### Speicherort:

Im Falle, dass das Plugin eine Konfigurationsanwendung zur Verfügung stellt, kann sich diese im <CodabixInstallDir>-Verzeichnis befinden. Die nachfolgende Liste zeigt die Hierarchie:

- <CodabixInstallDir>
  - <PluginConfigurationAppName>
  - ...

Zum Beispiel:

- C:\Program Files\Traeger\CoDaBix\
  - CoDaBix.S7DevicePlugin.Configurator.exe
  - ...

### Verwenden einer Konfigurationsdatei

## Speicherort

Falls das Plugin eine Konfigurationsdatei benutzt, kann diese im Pluginverzeichnis innerhalb des CoDaBix-Projektverzeichnisses (eingestellt in CoDaBix) liegen. Die nachfolgende Liste zeigt die Hierarchie:

- <CodabixProjectDir>
  - plugins\
    - <PluginConfigurationFileName>
    - ...

Zum Beispiel:

- D:\Data\Traeger\CoDaBix.Data\
  - CoDaBix.S7DevicePlugin.Settings.xml
  - [CoDaBix.S7DevicePlugin.Settings.xsd]
  - ...

## Struktur

Die XML-basierte Plugin-Konfigurationsdatei definiert mindestens das „PluginSettings“ Element, während die weiteren definierten Elemente von der Klassifizierung des Plugins und eigener benutzerdefinierter Elemente und Attribute abhängen.

```
<?xml version="1.0" encoding="utf-8" ?>
<PluginSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- Plugin specific child elements -->
</PluginSettings>
```

## Objektattribute

Einige Elemente in der Konfigurationsdatei erstellen eine Entität in CoDaBix® oder symbolisieren diese. Für solche Elemente sind bestimmte Attribute reserviert, um solche Entitäten leichter zu warten.

### Entität Identifier Attribut

Dieses Attribut ist vom Typ GUID. Dieser ID wird beim Erstellen der Entität von CoDaBix® erstellt. Durch diese Kennung (= in CoDaBix® der Global Node Identifier) ist es möglich, die Entität eindeutig zu identifizieren und dem Konfigurationselement eindeutig zuzuordnen.

Falls ein Konfigurationselement fehlt oder das mit dem ID zu identifizierende Element bzw. Attribut nicht gefunden wird, wird angenommen, dass eine neue Entität erstellt werden soll. Wenn eine neues Konfigurationselement erzeugt werden soll, ist es nicht nötig, das „Identifier“ Attribut zu setzen. Soll ein bestehendes Element geändert werden, ist es unerlässlich, das „Identifier“ Attribut zu setzen.

### Entität ChangeType Attribut

Die folgenden Werte gelten für alle Attribute dieser Art:

Wert	Beschreibung
„None“	Das Entitäts-Konfigurationselement wurde nicht rekursiv geändert.

Wert	Beschreibung
„Created“	Das Entitäts-Konfigurationselement ist neu und erfordert, dass eine neue Entität geschaffen werden muss.
„Updated“	Das Entitäts-Konfigurationselement wurde geändert und erfordert, dass die dargestellte Entität aktualisiert werden soll.
„Deleted“	Das Entitäts-Konfigurationselement ist veraltet und erfordert, dass die dargestellte Entität gelöscht werden soll.

Für den Fall, dass das Attribut den Wert hat:

- „None“: Keine besondere Aktion muss erfolgen und der Wert wird ignoriert. Das Plugin evaluiert anschließend das Attribut.
- „Created“: Das Identifier Attribut wird benutzt (falls gesetzt), um die Entität zu finden. Wird diese Entität nicht gefunden, wird eine neue Entität erstellt. Der Global Node Identifier der Entität wird dem Identifier Attribut hinzugefügt. Das Plugin evaluiert anschließend das Attribut.
- „Updated“: Das Identifier Attribut wird benutzt, um die Entität zu finden. Wird diese Entität nicht gefunden, wird eine neue Entität erstellt. Der übergebene Wert wird upgedatet. Das Plugin evaluiert anschließend das Attribut.
- „Deleted“: Das Identifier Attribut wird benutzt, um die Entität zu finden. Wird diese Entität gefunden, wird diese gelöscht. Nachfolgend überprüft das Plugin, ob das Attribute entfernt wurde.

## Synchronisation

Sobald ein Plugin durch den CoDaBix® Plugin Manager geladen und gestartet wurde, wird die Konfigurationsdatei (falls vorhanden) eingelesen und vom Plugin mit den entsprechenden CoDaBix® Entitäten synchronisiert.

Wird die Konfigurationsdatei geändert, informiert der Plugin Manager das entsprechende Plugin. Das Plugin entscheidet daraufhin, ob ein Neustart oder eine Synchronisation der entsprechenden Konfigurationselementen stattfindet.

Wird mindestens eine Konfigurationselement geändert, muss das Plugin das Entity Change Event behandeln und die Entität(en) mit der Konfigurationsdatei synchronisieren.

From:

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

Permanent link:

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

Last update: **2021/10/27 15:16**