

Inhaltsverzeichnis

Abkürzungen / Glossar

- **Codabix:** Communication Data Bridge. Dies ist der Name der Software, die dieses Handbuch behandelt.
- **Codabix Webkonfiguration:** Eine Webseite (HTML-Seite), über die Sie Codabix konfigurieren können, z.B. um die Nodes zu verwalten. Sie wird durch einen eingebetteten Webserver ausgeliefert und kann entweder über die Codabix-Anwendung (unter Windows) oder durch einen externen Browser (wie Mozilla Firefox, Microsoft Edge oder Google Chrome) aufgerufen werden.
- **Datenpunkt:** Siehe Node.
- **Projektverzeichnis:** Ein Verzeichnis auf der Festplatte, in dem die Codabix Backend-Datenbank (wenn die standardmäßig benutzte, eingebettete Datenbank verwendet wird), Plugin-Konfigurationsdateien und Protokolldateien abgelegt werden. Der Platzhalter <CodabixProjectDir> in den Plugin-Dokumentationen verweist auf dieses Verzeichnis.
- **Global Identifier:** Der Global Identifier eines Nodes ist eine 128-Bit-Ganzzahl (GUID, welche als hexadezimale Ziffern ausgedrückt wird), die einen Node in einer Codabix-Instanz eindeutig identifiziert. Der Global Identifier kann zum Synchronisieren von Nodes über verteilte Codabix-Instanzen verwendet werden. Wenn ein neuer Node erstellt wird, bekommt dieser entweder einen neuen Global Identifier, welcher über einen Zufallszahlengenerator generiert wird (wodurch die Wahrscheinlichkeit für einen doppelten Identifier selbst über verteilte Maschinen hinweg sehr gering ist), oder er bekommt eine bereits existierende GUID, wenn der Node zum Synchronisieren eines gleichen Nodes auf einer anderen Maschine verwendet werden soll.
- **Local Identifier:** Der Local Identifier eines Nodes ist eine 64-Bit-Ganzzahl, die einen Node innerhalb einer Codabix-Instanz eindeutig identifiziert. Im Gegensatz zu einem Global Identifier kann ein Local Identifier nicht zum Synchronisieren von Nodes über verteilte Codabix®-Instanzen auf anderen Maschinen benutzt werden, da diese seriell generiert werden (wodurch verschiedene Nodes auf unterschiedlichen Maschinen den gleichen Local Identifier bekommen können).
- **Node:** Ein Node (Knoten) ist ein fundamentales Objekt in Codabix, welches entweder ein **Datenpunkt-Node** ist, der Werte speichern kann, oder ein **Folder-Node** (Ordner-Node), welcher Unterknoten enthalten kann. Nodes werden in einem Baum (Tree) organisiert und können sowohl über die **Codabix Webkonfiguration** als auch über das REST-Interface und von Plugins bzw. Geräten aus konfiguriert werden. Jeder Node wird eindeutig über einen **Local Identifier** (ID, 64-Bit Ganzzahl) und einen **Global Identifier** (GUID, 128-Bit) identifiziert. Zusätzlich hat jeder Node einen Namen, welcher den Node eindeutig innerhalb seines Parent-Nodes identifiziert. Sie können die Local und Global Identifiers eines Nodes über die Codabix Webkonfiguration durch den „Access“-Button auslesen.

- **Node-Path:** Ein Node-Path ist eine Zeichenfolge, die angibt, wie zu einem bestimmten Node, ausgehend vom Root-Node, navigiert wird; ähnlich wie ein Pfad im Dateisystem. Ein Beispiel eines Nodepfads: `"/Nodes/Demo-Nodes/Temperature"`
Sie können den Pfad zu einem Node über die Codabix Webkonfiguration über den „Access“-Toolbar-Button auslesen.
Hinweis: Ein Node hat ebenfalls eine **Path**-Eigenschaft. Dies ist jedoch nicht dasselbe wie der **Node-Path** - der „Path“ eines Nodes enthält ein plugin-spezifisches Adressierungsschema, z.B. für S7-Variablen ist dies die S7-Adresse.
- **Plugin:** Eine funktionale Erweiterung von Codabix, die als Binärdateien (.DLL) ausgeliefert wird. Mit Plugins können neue Interfaces und Devices zu Codabix hinzugefügt werden, ohne das Codabix-Hauptprogramm ändern zu müssen. Ein Plugin wird im Codabix-Programmverzeichnis abgelegt; daher wird es durch die Codabix-Installation bereitgestellt, nicht durch das gewählte Projekt.
- **REST Interface:** Ein Interface zu Codabix über das HTTP(S)-Protokoll. Dies ermöglicht das Browsen, Lesen und Schreiben von Nodes durch das Senden von HTTP-POST-Anfragen mit einem JSON-Inhalt (Payload).
Wie auch die Codabix Webkonfiguration wird das REST-Interface durch den eingebetteten Webserver ausgeliefert.
- **Script:** Eine leichtgewichtige Erweiterung von Codabix, die in JavaScript/TypeScript geschrieben ist und dynamisch über den **Script-Editor** in der Codabix Webkonfiguration in Codabix integriert werden kann. Durch Scripts ist ein einfaches Programmieren von Codabix möglich. Diese werden in einer isolierten Umgebung ausgeführt (sie können nur definierte Codabix-APIs wie das Managen von Nodes aufrufen, jedoch keine Betriebssystem-APIs) und ihr Programmcode kann verändert werden, während Codabix gerade läuft.
Im Gegensatz zu einem normalen Plugin wird ein Script als Quellcode in der Codabix-Backend-Datenbank gespeichert und ist deshalb an das Projekt gebunden.
- **Script-Editor:** Ein JavaScript/TypeScript-Editor innerhalb der Codabix Webkonfiguration, der eine Vielzahl an Entwicklerfeatures (wie IntelliSense) bietet, wodurch das Schreiben von **Scripts** leicht gemacht wird.
- **Subscription:** Eine Aufforderung an ein Gerät, bestimmte Variablen regelmäßig zu lesen und die gelesenen Werte entweder nur bei Änderung oder nach jedem Lesevorgang in die entsprechenden Datenpunktnodes zu schreiben.
- **Synchroner Lesevorgang** (auch Synchronous Read): Eine Operation, die ein Gerät dazu auffordert, bestimmte Variablen tatsächlich zu lesen und die gelesenen Werte in den dazugehörigen Codabix-Nodes abzulegen. Es wird garantiert, dass die Variablen **nach** dem Zeitpunkt des Aufrufs des synchronen Lesevorgangs physisch gelesen werden.

From:

<https://codabix.de/> - **CoDaBix®**

Permanent link:

<https://codabix.de/de/abbreviations>

Last update: **2023/07/28 13:03**