

Table of Contents

Abbreviations / Glossary

- **CoDaBix®:** Communication Data Bridge. This is the name of the software which this documentation covers.
- **CoDaBix® Web Configuration:** A webpage (HTML page) that allows to configure CoDaBix®, e.g. managing its Nodes. It is served by an embedded web Server and can either be accessed using the CoDaBix® program, or by using an external web browser (like Mozilla Firefox or Google Chrome).
- **Datapoint:** See Node.
- **Data Directory:** A directory on the file system which stores the CoDaBix® back-end database (when using the default, embedded database), plugin configuration files and log files.
The Placeholder <CodabixDataDir> in the plugin documentation refers to this directory.
- **Global Identifier:** A Global Identifier of a Node is a 128-Bit Integer (GUID) expressed as hexadecimal digits that uniquely identifies a Node in a CoDaBix® instance and can be used to synchronize Nodes across distributed CoDaBix® instances. When a Node is created, it gets either a new Global Identifier created with a random number generator (which means the probability of duplicate identifiers is extremely low, even across distributed machines), or gets an existing GUID (if the Node is to be synchronized with an equal Node on another machine).
- **Local Identifier:** A Local Identifier of a Node is a 64-Bit Signed Integer that uniquely identifies a Node in a CoDaBix® instance. In contrast to a Global Identifier, Local Identifiers cannot be used to synchronize Nodes across distributed CoDaBix® instances on other machines because they are generated serially (so different Nodes may get the same Local Identifier on different machines).
- **Node:** A Node is a fundamental object in CoDaBix® that is either a **Datapoint Node** which can store a value, or a **Folder Node** which can contain child Nodes. Nodes are organized in a tree and can be configured and accessed with the **CoDaBix® Web Configuration** and additionally by the REST/JSON interface and from plugins / devices. Each Node is uniquely identified by a **Local Identifier** (ID, 64-Bit Integer) and a **Global Identifier** (GUID, 128-Bit). Additionally, each Node has a name that uniquely identifies this Node within its parent Node. You can retrieve the local and global identifiers of a Node in the CoDaBix® Web Configuration using the "Access" toolbar button.
- **Node Path:** A Node path is a string that says how to navigate to a specific Node from the root Node by using the Node names, similar to a path in the file system. An example of a path is "/Nodes/Demo-Nodes/Temperature".
You can retrieve the path to a specific Node in the CoDaBix® Web Configuration by using the "Access" toolbar button.
Note: A Node also has a "Path" property. This is however not the same as the **Node**

Path - the "Path" of a Node refers to a plugin specific addressing scheme, e.g. for S7 variables it is the PLC address.

- **Plugin:** An extension in functionality to CoDaBix®, delivered as binary files (.DLL). With plugins, new interfaces and devices can be added to CoDaBix® without having to change the main CoDaBix® program.
A plugin is stored in the CoDaBix® program directory; therefore, it is not bound to the selected data directory.
- **REST/JSON Interface:** An interface to CoDaBix® using the HTTP protocol. This allows to browse, read, and write Nodes by sending HTTP POST requests using a JSON payload. Just like the CoDaBix® Web Configuration, it is served by the embedded web Server of CoDaBix®.
- **Script:** A lightweight CoDaBix® extension, written in JavaScript / TypeScript and added dynamically to CoDaBix® via the **Script Editor** in the CoDaBix® Web Configuration. Script Plugins provide a simple way to program CoDaBix®. They are executed in an isolated environment (they can only access defined CoDaBix® APIs like managing Nodes, not OS APIs) and their program code can be changed while CoDaBix® is running. Unlike a regular plugin, a Script Plugin is stored as source code in the CoDaBix® database and therefore bound to the data directory.
- **Script Editor:** A JavaScript/TypeScript editor within the CoDaBix® Web Configuration with a rich developer experience (like IntelliSense) which makes writing **Scripts** a joy.
- **Subscription:** A request to a device to read specific variables in a regular interval and write the read values to the corresponding datapoint Nodes, either only if the value has changed, or every time they have been read.
- **Synchronous Read:** An operation that requests a device to actually read specific variables and store the values in the corresponding datapoint Nodes. The values are guaranteed to be physically read **after** the point in time when the synchronous read operation has been initiated.

From:

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

Permanent link:

<https://www.codabix.com/en/abbreviations>

Last update: **2019/05/29 16:01**