

# Table of Contents

<b>General</b>	1
<b>What Does the Plugin Do?</b>	1
<b>Features</b>	1
<b>Supported Database Engines</b>	1
<b>Purpose &amp; Use Cases</b>	2
<b>Installation</b>	2
Requirements	2
<b>Configuration</b>	2
Overview	2
Database-Specific Settings	3
Table-Specific Settings	5
Columns	5
<b>Diagnostics</b>	6
Database	6
Columns	7
Log File	7
<b>Entities</b>	7
Exchange	7
Channel	8
Table	8
Column	8
<b>Folders &amp; Files</b>	8
Folders	8
Files	9
<b>About Versions</b>	9
This Document	9
Plugin	9
Assembly	9



# SQL Exchange Plugin

## General

The SQL Exchange Plugin allows you to bidirectionally exchange CoDaBix® datapoint Node values with external databases, such as MySQL, Oracle or Microsoft SQL Server.

## What Does the Plugin Do?

The plugin allows you to:

- define databases (connections)
- browse or define tables
- browse or define table columns
- read and write columns, specifying a Read or Write SQL Expression

## Features

- Read columns of a database row (selected by the *Read SQL Expression*)
- Write columns of a database row by either inserting a new row or updating an existing row (selected by the *Write SQL Expression*)
- Subscribe to columns of a database row when the corresponding CoDaBix® variable Nodes are subscribed to

## Supported Database Engines

- MySQL 5.5 or higher
- Microsoft SQL Server 2008 or higher
- Oracle
- SQLite

## Purpose & Use Cases

- Append a table row or update an existing row
- Read data from a table from the latest row or from a specific row
- Create subscriptions to get notified when data in the database is changed
- Call **stored procedures** using method nodes in Codabix (currently only supported for **Oracle** databases)

## Installation

This plugin is part of the CoDaBix® Setup. Please consult [CoDaBix® Setup and First Start](#) for more information on how to install and uninstall this plugin.

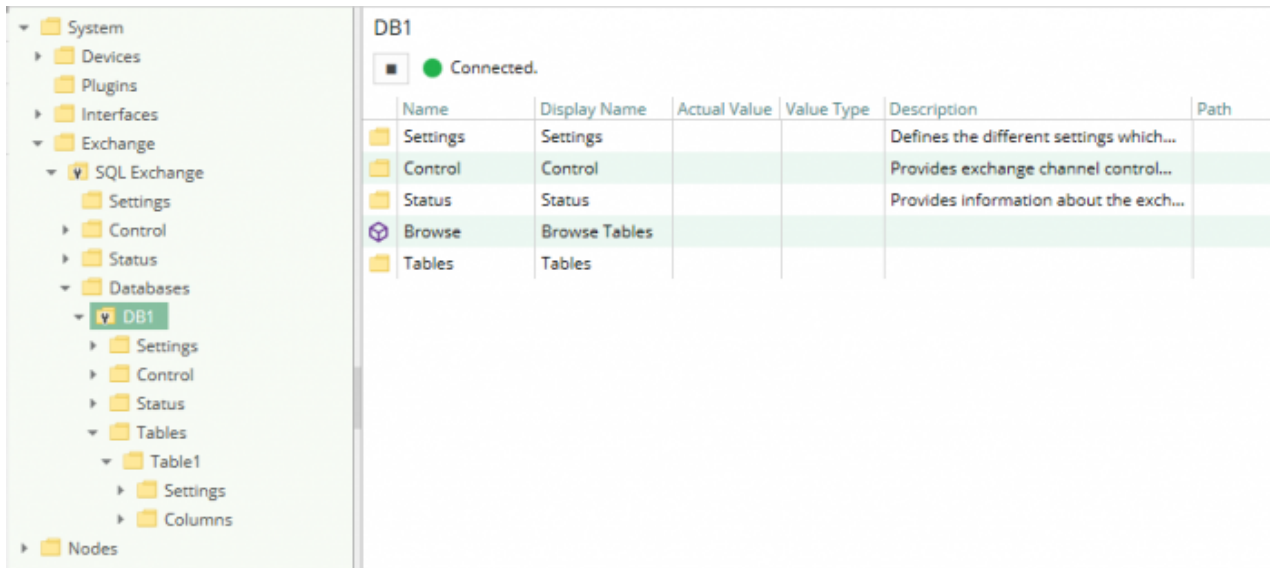
### Requirements

- The machine which runs CoDaBix® must have access to one of the supported database engines.

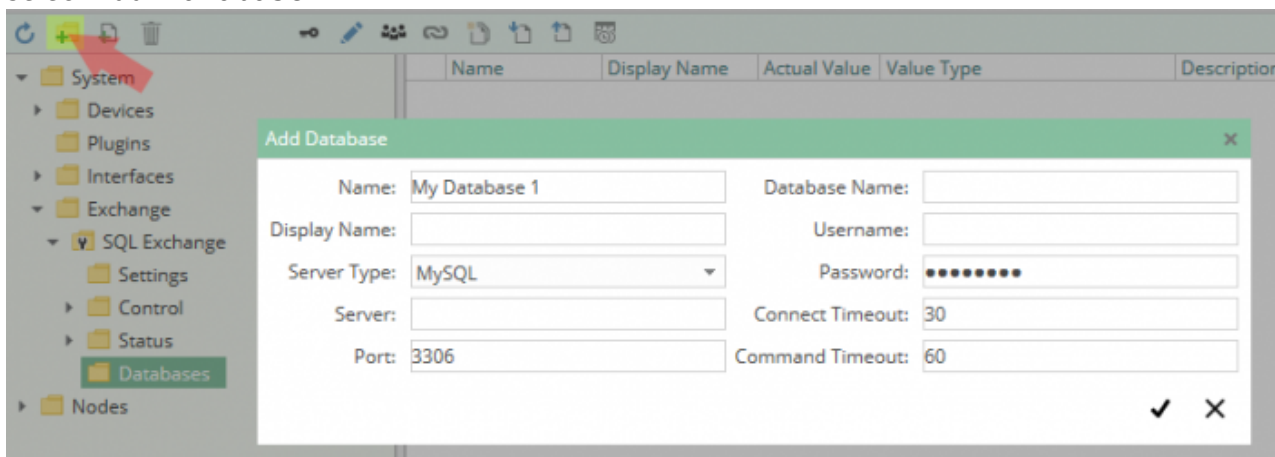
## Configuration

### Overview

The entire SQL Exchange Plugin configuration is located under the Node path /System/Exchange/SQL Exchange.



The Node tree in the image above depicts the SQL Exchange Plugin's default Node tree. To set up one or more SQL Exchange **Databases**, add a Folder Node beneath the Node SQL Exchange / Databases, or right-click on the SQL Exchange / Databases Node and select Add Database.

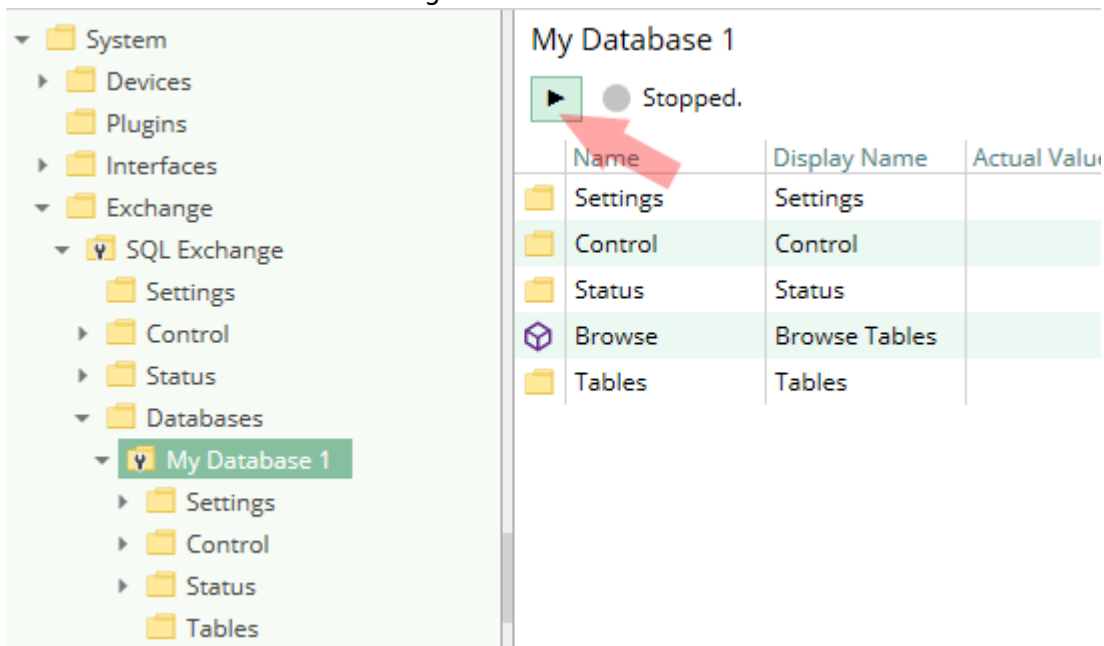


### Database-Specific Settings

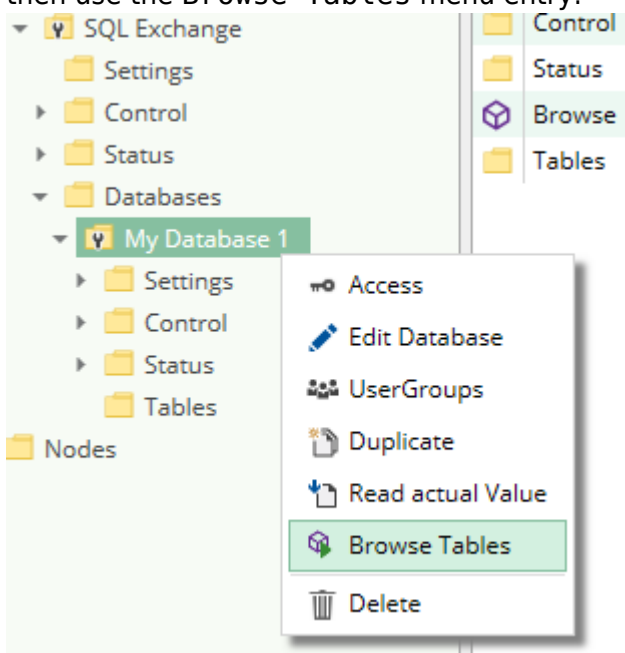
Name	Type	Description
Server Type	Enum	Defines the server type to which the database connection is established. Valid values: MySQL, Oracle, MSSQL, ODBC, SQLite
Server	String	The hostname or IP address to which the connection shall be established. For <b>SQLite</b> , this is the file path to the database; other connection properties like Port will be ignored.
Port	Integer	The port to which the connection shall be established. When using MSSQL as Server Type, you can specify 0 as port, which means a connection to the default SQL Server instance shall be established.
Database Name	String	The name of the database/scheme. For <b>ODBC</b> , this is the data source name (DSN), and is the only field that needs to be specified. The DSN can be configured on Windows using the <b>ODBC Data Sources</b> app (User DSN or System DSN). Note that for Codabix (x86) you will need to use the 32-bit and for Codabix (x64) the 64-bit ODBC Data Sources app.

Name	Type	Description
Username	String	The username for the connection.
Password	Password	The password.
Connect Timeout	Integer	The timeout in seconds after an attempt to connect is canceled.
Command Timeout	Integer	The timeout in seconds after a long-running command is canceled.

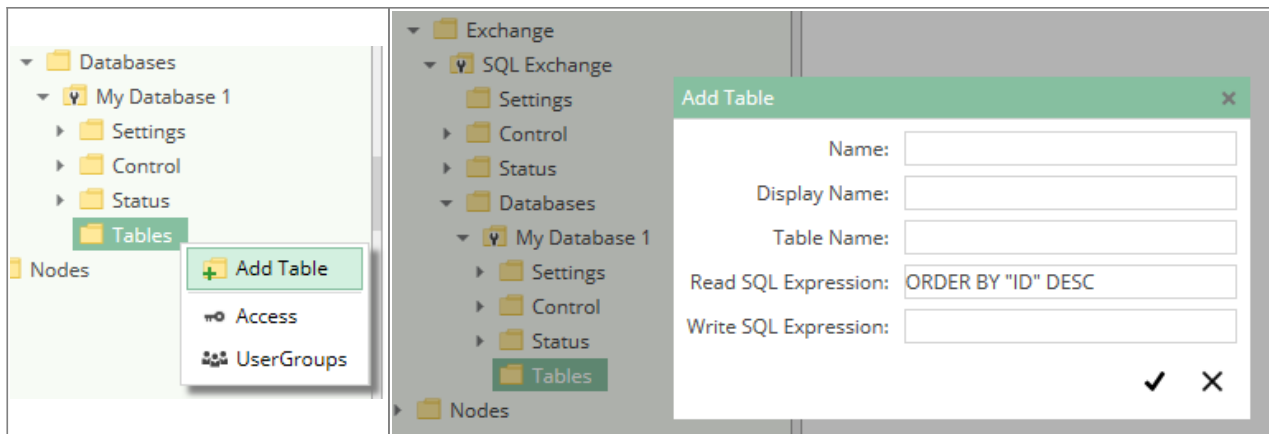
After clicking “Save”, the database Node will be created. You can start it by selecting the database Node and then clicking the Start button:



To setup one or more **tables** for the database, you can right-click on the database Node and then use the Browse Tables menu entry:



Alternatively, you can right-click on the Tables Node and select Add Table (to edit an existing table, right-click on the Table Node and select Edit Table):



### Table-Specific Settings

Name	Type	Description
Table Name	String	The name of the table in the database.
Read SQL Expression	String	An SQL string (e.g. a WHERE clause or an ORDER BY clause) that is applied when reading the columns. The first row returned by the database is read. You can specify a WHERE clause (e.g. WHERE "ID" = 123) to read only a specific column. The default is ORDER BY "ID" DESC to order the rows by ID descending, so that the row with the highest ID is used.
Write SQL Expression	String	An SQL String (e.g. a WHERE clause) that is applied when writing the columns. You can specify a WHERE clause (e.g. WHERE "ID" = 123) to update an existing row. The default value is an empty string, which means a new row is inserted every time column Nodes are written.

### Columns

Every Node within the Columns Node of a table Node can be assigned to a column of the table, using the Path property.

Name	Display Name	Actual Value	Value Type	Description	Path	Status
ID	ID	9	Int64		ID	Good
SpalteZahl	SpalteZahl	706	Int32		SpalteZahl	Good
SpalteText	SpalteText	Neu-für-706	String		SpalteText	Good
SpalteSyswert	SpalteSyswert		String		SpalteSyswert	Bad: Column is NULL
SpalteSyswert2	SpalteSyswert2		String		SpalteSyswert2	Bad: Column is NULL

### Syntax:

<ColumnName>	Only the column name is specified. When reading or writing from / to it, the table's Read SQL or Write SQL expressions are used. Example: MyColumn1
--------------	--

<ColumnName>; <ReadWriteExpression>	The column name is specified, along with an expression that is used when reading or writing the column. Example: MyColumn1; WHERE ID = 123
<ColumnName>; <ReadExpression>; <WriteExpression>	The column name is specified, along with an expression that is used when reading from the column, and a separate expression that is used when writing to the column. Example: MyColumn1; WHERE ID = 123; WHERE ID = 456

## Diagnostics

The SQL Exchange Plugin provides different status information depending on the layer to inspect. In general the channel-based diagnostic information is produced by the connection status of the channel to the database Server. The variable-based diagnostic information is produced during the read/write access of the different columns.

### Database

To monitor and diagnose the status of the different databases channels, take a look at the following image:




Name	Display Name	Actual Value	Value Type	Description
Settings	Settings			Defines th...
Control	Control			Provides...
Status	Status			Provides i...
Browse	Browse Tables			
Tables	Tables			

The image above depicts the database channel's Control Panel which displays all status relevant information. The control panel will automatically update its status information when a new status is available.

### Status Circle

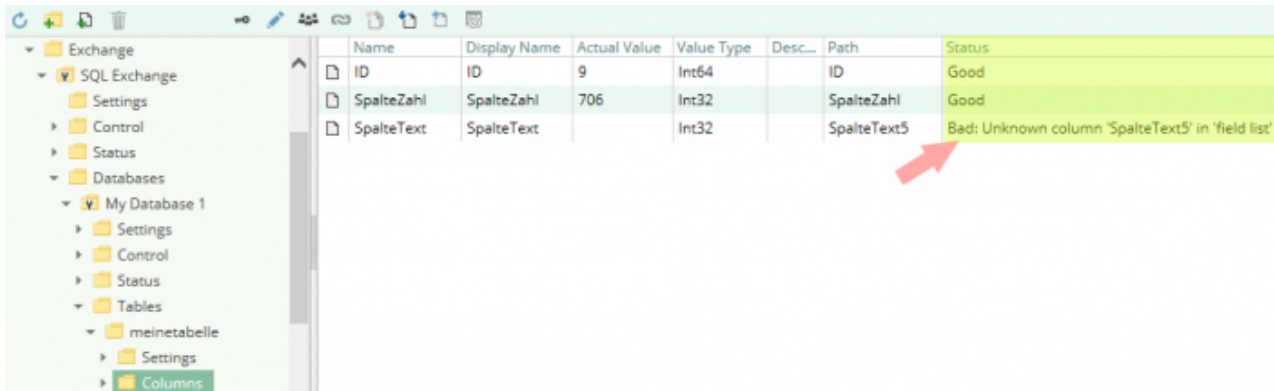
Color	Meaning
	The database channel is stopped. Click the ► button to start it.
	The database channel is currently in the progress of starting or stopping or is waiting to establish a connection.



Color	Meaning
	The database channel is running and a connection has successfully been established. You can stop it by clicking the  button.
	The database channel is running, but the connection is currently in an error state. Please check the status text for more information.

## Columns

To monitor and diagnose the status of the different table columns, take a look at the columns' Status property displayed in CoDaBix. Use the button "Read actual Value" to read the values from the database and store the result into the column nodes.



Name	Display Name	Actual Value	Value Type	Desc...	Path	Status
ID	ID	9	Int64		ID	Good
SpalteZahl	SpalteZahl	706	Int32		SpalteZahl	Good
SpalteText	SpalteText		Int32		SpalteText5	Bad: Unknown column 'SpalteText5' in 'field list'

## Log File

All database related status information is also logged into the database-specific log file stored in the [LoggingFolder]. Each log file is named in the naming scheme SQL Exchange.<DatabaseName>.log. The content of such a log file can look as follows:

```
...
[14:55:34 25.07.2017] - Error (Severity=High): Code=[-1], Text=[Access
denied for user 'abc'@'localhost' (using password: YES)], Details=[]
...
```

# Entities

As each exchange plugin the SQL exchange plugin extends the basic CoDaBix [Exchange Model](#).

## Exchange

The plugin's exchange type `SqlExchange` also defines the `SqlExchangeChannel` and therefore extends the basic `CodabixExchange` and `CodabixExchangeChannel` entities. While the `SqlExchange` just represents a concretization of the `CodabixExchange`, the `SqlExchangeChannel` extends the `CodabixExchangeChannel` with the SQL Table entities.

## Channel

Each channel is handled by a channel worker which establishes a physical connection to the database. For diagnostic purposes, the worker automatically checks the database connection every 10 seconds to update the status code of the Channel and description to detect connection failures.

By default, the worker does not read any values. When a Client or plugin requests a synchronous read of the Channel, the channel worker reads the variables in CoDaBix (e.g. using the CoDaBix Web Configuration's function "Read actual value") from the database and then writes them into the corresponding CoDaBix Nodes.

Similarly, when a Client or plugin writes values into the channel's variables, the channel worker will write those values to the database.

To have an database variable being read steadily, you can edit the Node in the Configuration Web GUI and set "History Options" to Yes (which will create an internal subscription), or you can use e.g. an OPC UA Client connected to the OPC UA Server plugin and create a subscription for the S7 variable Nodes. In these cases, the channel worker reads the variables from the database at a regular interval and, if the value of one of the variables has changed, writes the new value into the corresponding CoDaBix® Node.

## Table

Each table entity represents the information needed to access a database table. At table level, you can specify how to select the row which the values will be read from or written to.

## Column

Each column entity represents a single database table column.

# Folders & Files

## Folders

Content	Path	Usage
AssemblyFolder	<CodabixInstallDir>/plugins/SqlExchangePlugin/	Contains the plugin's assembly file.
ConfigFolder	<CodabixDataDir>/plugins/SqlExchangePlugin/	Contains the plugin's configuration file.
LoggingFolder	<CodabixDataDir>/log/	Contains the plugin's log files.

## Files

Type	Path	Usage
Assembly	[AssemblyFolder]/CoDaBix.SqlExchangePlugin.dll	The plugin's assembly file.
Logging	[LoggingFolder]/SQL Exchange.<DatabaseName>.log	The log file.

## About Versions

### This Document

<b>Date</b>	2018-06-15
<b>Version</b>	1.1

### Plugin

<b>Name</b>	SQL Exchange Plugin
<b>Node</b>	/System/Exchange/SQL Exchange
<b>Version</b>	1.0.5

### Assembly

<b>Name</b>	CoDaBix.SqlExchangePlugin.dll
<b>Date</b>	2019-02-04
<b>Version</b>	1.0.5.0

From:

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

Permanent link:

<https://www.codabix.com/en/plugins/exchange/sqlexchangeplugin>

Last update: **2021/07/30 13:40**