

Table of Contents

General	1
What Does the Plugin Do?	1
Features	1
Supported Contents	1
Purpose & Use Cases	2
Purpose	2
Use Cases	2
Installation	2
Requirements	2
Configuration	2
Using the Configuration File	2
Structure	2
Servers Element	3
Server Element	3
Trigger-Element	4
File Element	5
Bindings Element	7
Binding Element	7
Node Query Expression	8
Usage	9
Synchronization	9
Example Configuration File	9
Example Configuration Scheme File	10
Diagnostics	14
Entities	15
Folders & Files	15
Folders	15
Files	15
About Versions	15
This Document	15
Plugin	15
Assembly	16

CSV Exchange Plugin

General

The CSV Exchange Plugin provides a data exchange mechanism between CoDaBix® and CSV (comma-separated values) based file formats.

What Does the Plugin Do?

The CSV Exchange Plugin defines a storage structure for CSV files. Within this structure one or more data sets can be defined. Each data set represents a binding between the columns in the CSV storage and the Nodes in CoDaBix®. Through that binding the plugin reads and writes the CoDaBix® Node values as a data set to the CSV file and vice versa.

Features

- File change monitoring
- File truncation on completion
- File deletion on completion
- Custom CSV separator configuration
- Relative Node binding in DataSet
- Access files in the local file system or use an SSH-based transfer protocol (SCP or SFTP)

Supported Contents

- Microsoft Excel CSV Separator (default)
- Custom CSV separators

Purpose & Use Cases

Purpose

The configured CSV files can be used simply to read and write data from and to CoDaBix®. This allows simple data import/export scenarios using comma-separated values. Also other CoDaBix® participants can deliver and receive information stored in a CSV file through the CSV Exchange Plugin.

Use Cases

- Automatic data synchronization between CoDaBix® and third party (sub)systems to transfer data
- Data import to archive process protocols
- Data export to post-process CoDaBix®-produced and -stored information

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

- Basic requirements of CoDaBix®
- Read/write access to the CSV files configured

Configuration

Using the Configuration File

Structure

The CSV Exchange Plugin defines the root of its element tree by the `PluginSettings` element as described in [Plugin Configuration - Using a Configuration File](#) and continues its XML tree using the `Servers` element.

Servers Element

The Servers element serves as the container for one or more Server elements. This element maintains all file Servers associated with the exchange plugin.

The Servers element can look as follows:

```
<Servers>
  <!-- 0-n Server elements -->
</Servers>
```

Server Element

The Server element serves as the container for the element File. One server identifies one CSV file to which the Server configuration belongs. The configuration also includes the actions to perform before, during and after the synchronization progress. This information is therefore used by the CSV Exchange Plugin to exchange CSV data with CoDaBix®.

Each Server element provides the following list of attributes:

	Mandatory	Type	Purpose
IsEnabled	no	Boolean	Shows the value <code>true</code> if the data exchange is active for the file Server; otherwise shows the value <code>false</code> (default: <code>true</code>).
SyncMode	no	CSV Sync Mode	The direction into which the synchronization is to be performed.
Additional attributes for SyncMode="FileToSystem"			
SyncTrigger	no	CSV Sync Trigger	The trigger condition to initiate the synchronization.
BeforeSyncAction	no	CSV Sync Action	The action to perform, before the synchronization progress begins.
AfterSyncAction	no	CSV Sync Action	The action to perform, after the synchronization progress has ended.
BeforeSyncMoveFileTo (if BeforeSyncAction="MoveFile")	yes	Node Query Expression	Specifies the new fully qualified file path or directory path to which the file is moved. If you specify a directory path (ending with <code>/</code> or <code>\</code>), the file name is kept when moving the file to the new directory. In that case, if a file with the same name already exists, the plugin will append a suffix like <code>(1)</code> , <code>(2)</code> etc.
AfterSyncMoveFileTo (if AfterSyncAction="MoveFile")	yes	Node Query Expression	See above.

CSV Sync Mode

The following values are valid for attributes of that type:

Value	Description
"FileToSystem"	The CSV file is synchronized to the bound CoDaBix® Nodes.
"SystemToFile"	The bound CoDaBix® Nodes are synchronized to the CSV file.

CSV Sync Trigger

The following values are valid for attributes of that type:

Value	Description
"FileChanged"	The synchronization progress is to be triggered whenever the file gets changed. This means that the synchronization is invoked everytime as soon as the date changes at which the file was last written.

CSV Sync Action

The following values are valid for attributes of that type:

Value	Description
"KeepFile"	The file is to be kept before/after synchronization.
"TruncateFile"	The files content is to be removed before/after synchronization.
"DeleteFile"	The file is to be deleted before/after synchronization.
"MoveFile"	The file is to be moved or renamed before/after synchronization. For this action, you will have to specify the BeforeSyncMoveFileTo or AfterSyncMoveFileTo attributes (see above).

The Server element can look as follows:

```

<Server SyncMode="FileToSystem"
        SyncTrigger="FileChanged"
        AfterSyncAction="DeleteFile">
  <!-- 0-n Trigger elements -->
  <!-- File element -->
</Server>

```

Trigger-Element

The Trigger element allows to define a trigger (if SyncMode="SystemToFile"), which specifies when the data should be collected from CoDaBix® by doing a synchronous read and then be written into the CSV file.

The Trigger element provides the following list of attributes:

	Mandatory	Type	Purpose
Type	yes	String	Specifies the type of the trigger. "ValueChange": The trigger fires when a new value is written to a specified node and the new value is different from the previous value of the node. "Edge": The trigger fires when a specific values is written to the node (specified by the EdgeValue attribute) and the previous value was different. Optionally, you can specify a value that is to be written back to the node after synchronization. "Interval": The trigger fires in a regular interval (specified by the interval attribute).
Node (if Type="Edge" or Type="ValueChange")	yes	String	Specifies the fully qualified path to the node which is to be monitored.
EdgeValue (if Type="Edge")	yes	String	Specifies the value for which the trigger is monitoring. The trigger fires when this value is written to the node while the previous value of the node was a different one.
ChangeBackValue (if Type="Edge")	no	String	If specified, the CSV Exchange plugin will write this value into the node after the trigger fired and the dataset has been collected with an synchronous read.
ChangeBackNode (if Type="Edge")	no	String	If specified and if ChangeBackValue is specified, the CSV Exchange Plugin will write the value to the node specified by this attribute, instead of the node specified by the Node attribute.
Interval (if Type="Interval")	yes	Int32	Specifies the trigger interval in milliseconds. For example, a value of 2000 means that the trigger fires every two seconds.


The Trigger element can look as follows:

```
<Trigger Type="Edge" Node="/Nodes/Line1/TriggerNode" EdgeValue="1"
ChangeBackValue="0" />
```

File Element

The File element serves as the container for the element Bindings and defines the attributes to set up the file used for the exchange. These attributes configure the file and its format used to synchronize.

The File element provides the following list of attributes:

	Mandatory	Type	Purpose
Path	yes	Node Query Expression	<p>The fully qualified accessible path to the CSV file to synchronize.</p> <p>When using SyncMode SystemToFile, the nodes specified in the Node Query Expression will be included in the synchronous read of the column nodes, and then evaluated when writing the file.</p> <p>When using SyncMode FileToSystem, the specified nodes will be evaluated once when starting the channel, and additionally you can specify a filter like XY*.csv, which means the plugin will monitor and process all files in the directory matching that filter.</p> <p>Note: For files accessed using the local file system, this path is subject to the File Access Security restrictions that have been defined in the CoDaBix® Settings.</p>
Separator	no	Char	The CSV separator in the file used to separate the values from each other (default: ;).
HasHeader	no	Bool	<p>Specifies if the first line of the CSV file contains a title for each column. Valid values: True, False</p> <p>When specifying True, for SyncMode FileToSystem this means that the first line of the CSV file will be skipped when synchronizing it. For SyncMode SystemToFile, this means that if the CSV file is empty when synchronizing it, a header line will first be written with a title for each column that can be specified in the <Binding> elements using the HeaderName attribute.</p>
Type	no	String	<p>Specifies the type of the file transfer mechanism. File (default): The file is accessed using the local file system. Scp: The file is accessed via an SSH Server (SCP protocol). Sftp: The file is accessed via an SSH Server (SFTP protocol).</p>
Additional attributes, when Type="Scp" or Type="Sftp" :			
Hostname	yes	String	The hostname of the SSH Server.
Username	yes	String	The username for the SSH Server.
Password	yes	String	<p>The encrypted password for the SSH Server. To get the encrypted password, open CoDaBix, and in the Web Configuration click on  Password Security. Here, you can enter your original password and encrypt it, so that it can be specified for this attribute.</p>

The File element can look as follows:

```
<File Path="MachineSetup.csv">
  <!-- Bindings element -->
```



```
</File>
```

or, when using SCP:

```
<File Path="/directory/MachineSetup.csv" Type="Scp"
Hostname="192.168.0.20" Username="user1" Password="password1">
  <!-- Bindings element -->
</File>
```

Bindings Element

The Bindings element serves as the container for one or more Binding elements. This element maintains all bindings associated with the file.

The Bindings element provides the following list of attributes:

	Mandatory	Type	Purpose
BaseNode	no	String	The Node path used as the base Node path for all subsequent Node bindings.

The Bindings element can look as follows:

```
<Bindings BaseNode="/Nodes/Line 1/Tools/CuttingTool">
  <!-- 0-n Binding elements -->
</Bindings>
```

Binding Element

The Binding element binds a column in the CSV file to a CoDaBix® Node. This information is therefore used by the CSV Exchange Plugin to link the CSV comma-separated values to CoDaBix® Nodes.

Each Binding element provides the following list of attributes:

	Mandatory	Type	Purpose
ColumnIndex	yes	Int32	The zero-based (index of the first column is zero) index of the CSV column to bind.
Node	yes (if SystemValueType is not specified)	String	The (relative, if BaseNode is used) Node path of the Node to bind.
ValueType (if Node is specified)	no	Value Type	If specified, forces the column value to be interpreted as the specified type. You can specify the formatting using the ValueFormat attribute.
ValueFormat (if ValueType is specified)	no	String	Specifies the formatting for the value.

	Mandatory	Type	Purpose
SystemValueType	yes (if Node is not specified)	System Value Type	If specified, instead of using the value of a node, a system value is used for this column. See the table below for the possible system value types.
SystemValueFormat (if SystemValueType is specified)	no	String	Specifies the format of the system value (e.g. date format).
HeaderName	no	Node Query Expression	When HasHeader is set to True on the <File> element, specifies the title of the column. If not specified, a default name like "Column1" will be used.

Value Type

The following values are valid for attributes of that type:

Value	Description
"DateTime"	The column value is to be interpreted as a DateTime value. You can specify the formatting using a .NET date/time format string (standard or custom), e.g. yyyy-MM-dd HH:mm:ss.
"TimeSpan"	The column value is to be interpreted as a TimeSpan value. You can specify the formatting using a .NET TimeSpan format string (standard or custom), e.g. dd\ .hh\:mm:ss.

System Value Type

The following values are valid for attributes of that type:

Value	Description
"TriggerTimestamp" (if SyncMode="SystemToFile")	The column will contain the timestamp when the trigger has fired. You can specify the formatting using a .NET date/time format string (standard or custom), e.g. yyyy-MM-dd HH:mm:ss.
"CreationTimestamp" (if SyncMode="FileToSystem")	The timestamp specified in the column will be used as CreationTimestamp for the node values that are to be written to CoDaBix. You can specify the formatting using a .NET date/time format string (standard or custom), e.g. yyyy-MM-dd HH:mm:ss.

The Binding element can look like as follows:

```
<Binding ColumnIndex="0" Node="Depth"/>
<Binding ColumnIndex="1" Node="Last Refresh" ValueType="DateTime"
ValueFormat="yyyyMMdd-HH:mm:ss"/>
<Binding ColumnIndex="2" SystemValueType="TriggerTimestamp"
SystemValueFormat="yyyyMMdd-HH:mm:ss"/>
```

Node Query Expression

Certain attributes (e.g. file path or column header) allow to specify a node path, which means that part will be replaced with the value of the node. A Query Expression has the form `${<Node-Path>|<ID>|<Guid>}` (the node path itself must not contain a `}` character). If you want to use a dollar sign (\$) directly, you have to escape it as two dollar signs (\$\$).

When evaluating an expression, a synchronous read is initiated. This allows you to provide the value with a script that registers a `codabix.NodeReader` for the specified node.

Examples (assuming node `/Nodes/A` has value "First", node `/Nodes/B` has value "Second"):

Node Query Expression	Resulting output/file path
<code>C:\File-\${/Nodes/A}.csv</code>	<code>C:\File-First.csv</code>
<code>\$\$Header_\${/Nodes/A}\${/Nodes/B}\$\$</code>	<code>\$Header_FirstSecond\$</code>

Usage

It is recommended to use a professional XML editor when editing the configuration file manually. To also benefit of the XML scheme definition mentioned at the end of this article you have to refer to the schema definition by using the `xsi:noNamespaceSchemaLocation` attribute in the document root `PluginSettings` as follows (the XSD file needs to be placed besides the XML file):

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

Independent from the technique of editing/creating the configuration file manually or automated the element tree documented above is to be fulfilled to produce a valid, well-formed and usable configuration file.

Synchronization

As soon as the plugin is loaded and started by the CoDaBix® Plugin Manager its configuration file is read and synchronized by the plugin with the appropriate CoDaBix® entities.

In case the configuration file changes the plugin manager notifies the plugin. Through that notification the plugin restarts and uses the latest configuration changes on startup.

Example Configuration File

[Codabix.CsvExchangePlugin.Settings.xml](#)

```
<PluginSettings
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Codabix.CsvExchangePlugin.Settings.xsd">
  <Servers>
    <Server SyncMode="FileToSystem"
            SyncTrigger="FileChanged"
```

```

        AfterSyncAction="MoveFile"
        AfterSyncMoveFileTo="ProcessedFiles\">
    <File Path="SampleFile*.csv">
        <Bindings BaseNode="/Nodes/Line 1/Tools/CuttingTool">
            <Binding ColumnIndex="0" Node="Depth"/>
            <Binding ColumnIndex="1" Node="Speed"/>
            <Binding ColumnIndex="2" Node="Direction/X"/>
            <Binding ColumnIndex="3" Node="Direction/Y"/>
            <Binding ColumnIndex="4" Node="Date"
ValueTypes="DateTime" ValueFormat="yyyyMMdd-HH:mm:ss"/>
        </Bindings>
    </File>
</Server>

    <Server SyncMode="SystemToFile">
        <Trigger Type="Edge" Node="/Nodes/Line
2/Feedback/TriggerNode" EdgeValue="1" ChangeBackValue="0" />
        <File Path="/home/user/${/Nodes/Line2/CsvFileName}"
Type="Sftp"
Hostname="192.168.0.20"
Username="user1"
Password="encrypted-password"
HasHeader="True">
            <Bindings BaseNode="/Nodes/Line 2/Feedback">
                <Binding ColumnIndex="0" Node="CurrentDepth"
HeaderName="My Column 1"/>
                <Binding ColumnIndex="1" Node="CurrentSpeed"
HeaderName="My Column 2"/>
                <Binding ColumnIndex="2"
SystemValueType="TriggerTimestamp" SystemValueFormat="yyyyMMdd-
HH:mm:ss" HeaderName="My Column 3"/>
            </Bindings>
        </File>
    </Server>
</Servers>
</PluginSettings>

```

Example Configuration Scheme File

Codabix.CsvExchangePlugin.Settings.xsd

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Codabix.CsvExchangePlugin.Settings"
    elementFormDefault="qualified"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="SystemValueType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="CreationTimestamp" />
            <xs:enumeration value="TriggerTimestamp" />

```

```

    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ValueTypeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Integer" />
      <xs:enumeration value="FloatingPoint" />
      <xs:enumeration value="DateTime" />
      <xs:enumeration value="TimeSpan" />
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="BindingType">
    <xs:attribute name="ColumnIndex" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="0" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="HeaderName" use="optional">
      <xs:simpleType>
        <xs:restriction base="xs:string" />
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Node" use="optional">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="1" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="SystemValueType"
type="SystemValueTypeType" use="optional" />
    <xs:attribute name="SystemValueFormat" type="xs:string"
use="optional" />
    <xs:attribute name="ValueType" type="ValueTypeType"
use="optional"/>
    <xs:attribute name="ValueFormat" type="xs:string"
use="optional" />
  </xs:complexType>

  <xs:complexType name="BindingsType">
    <xs:sequence>
      <xs:element name="Binding" type="BindingType" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>

    <xs:attribute name="BaseNode" type="xs:string" use="optional"
/>
  </xs:complexType>

```

```
<xs:simpleType name="FileTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="File" />
    <xs:enumeration value="Scp" />
    <xs:enumeration value="Sftp" />
    <xs:enumeration value="file" />
    <xs:enumeration value="scp" />
    <xs:enumeration value="sftp" />
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="HasHeaderType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="True" />
    <xs:enumeration value="False" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="FileType">
  <xs:sequence>
    <xs:element name="Bindings" type="BindingsType"
minOccurs="1" maxOccurs="1" />
  </xs:sequence>

  <xs:attribute name="Path" type="xs:string" use="required" />
  <xs:attribute name="Type" type="FileTypeType" use="optional"
/>
  <xs:attribute name="Hostname" type="xs:string" use="optional"
/>
  <xs:attribute name="Username" type="xs:string" use="optional"
/>
  <xs:attribute name="Password" type="xs:string" use="optional"
/>
  <xs:attribute name="MonitoringInterval" type="xs:integer"
use="optional" />

  <xs:attribute name="HasHeader" type="HasHeaderType"
use="optional" />

  <xs:attribute name="Separator" use="optional" default=";">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:length value="1" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:simpleType name="TriggerTypeType">
  <xs:restriction base="xs:string">
```

```

        <xs:enumeration value="Edge" />
        <xs:enumeration value="ValueChange" />
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="TriggerType">
    <xs:attribute name="Type" type="TriggerTypeType"
use="required" />
    <xs:attribute name="Node" type="xs:string" use="optional" />
    <xs:attribute name="EdgeValue" type="xs:string" use="optional"
/>
    <xs:attribute name="ChangeBackValue" type="xs:string"
use="optional" />
    <xs:attribute name="ChangeBackNode" type="xs:string"
use="optional" />
</xs:complexType>

<xs:simpleType name="SyncModeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="FileToSystem" />
        <xs:enumeration value="SystemToFile" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SyncTriggerType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="FileChanged" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="SyncActionType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="KeepFile" />
        <xs:enumeration value="TruncateFile" />
        <xs:enumeration value="DeleteFile" />
        <xs:enumeration value="MoveFile" />
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="ServerType" mixed="true">
    <xs:sequence>
        <xs:element name="Trigger" type="TriggerType" minOccurs="0"
maxOccurs="unbounded" />
        <xs:element name="File" type="FileType" minOccurs="1"
maxOccurs="1" />
    </xs:sequence>

    <xs:attribute name="IsEnabled" type="xs:boolean"
use="optional" />
    <xs:attribute name="SyncMode" type="SyncModeType"
use="optional" />

```

```
<xs:attribute name="SyncTrigger" type="SyncTriggerType"
use="optional" />
<xs:attribute name="BeforeSyncAction" type="SyncActionType"
use="optional" />
<xs:attribute name="AfterSyncAction" type="SyncActionType"
use="optional" />
<xs:attribute name="BeforeSyncMoveFileTo" type="xs:string"
use="optional" />
<xs:attribute name="AfterSyncMoveFileTo" type="xs:string"
use="optional" />
</xs:complexType>

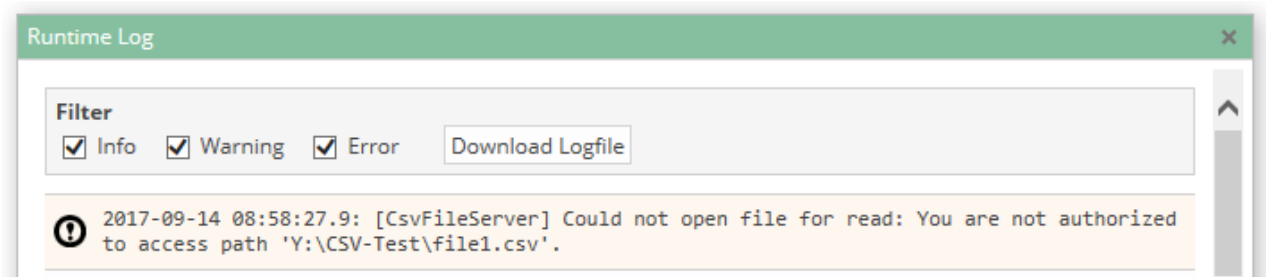
<xs:complexType name="ServersType">
  <xs:sequence>
    <xs:element name="Server" type="ServerType" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="PluginSettingsType">
  <xs:sequence>
    <xs:element name="Servers" type="ServersType" minOccurs="1"
maxOccurs="1" />
  </xs:sequence>
</xs:complexType>

<xs:element name="PluginSettings" type="PluginSettingsType" />
</xs:schema>
```

Diagnostics

The CSV Exchange Plugin will log events that start with [CsvFileServerPlugin] or [CsvFileServer] into the CoDaBix® runtime log. You can open the runtime log in the Web Configuration to view the last log entries:



Entities

The CSV Exchange Plugin does not use the CoDaBix® Entity Model because it configured by an XML Configuration File (Codabix.CsvExchangePlugin.Settings.xml) and therefore does not provide entities.

Folders & Files

Folders

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

Files

Type	Path	Usage
Assembly	[AssemblyFolder]/Codabix.CsvExchangePlugin.dll	The plugin's assembly file.
Config File	[ConfigFolder]/Codabix.CsvExchangePlugin.Settings.xml	The config file.

About Versions

This Document

Date	2019-04-17
Version	1.4

Plugin

Name	CSV File Server
Node	/System/Plugins/Exchange/CSV
Version	1.1.0

Assembly

Name	Codabix.CsvExchangePlugin.dll
Date	2019-04-16
Version	1.1.0.0

From:

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

Permanent link:

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

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