

Systems - CSV: Source of identities



OBSOLETE from IdM version 10.6.0. [Systems - CSV: Source of identities via wizard](#)

Introduction

This tutorial will show you how to connect csv files as a source of data about users in CzechIdM. We will use the default CSVConnector.

Data source

Our sample source file has just three columns which are login, firstname and lastname.

```
LOGIN, FIRSTNAME, LASTNAME
edwardw, Edward, Williams
adaml, Adam, Lewis
```

Move your source file onto the server on which IdM is running. It's a good practice to put CSV files of every project into an "import_csv" folder, which you can create in the /opt/ directory. Set the owner of the folder and the CSV files to tomcat:tomcat. Set permissions so that the CzechIdM server will be able to read this file. **Also, CzechIdM must be able to write into the folder where the file is located because the connector needs to create temporary files when processing csv into its location.** (chmod 750 for folder permissions)

We recommend using the column names as upper-case, because the connector is case-sensitive and also it transforms the names of the columns to uppercase automatically. If you don't use upper case, you would have to change the attributes in the Scheme manually (see later).

Basic information

Go to the **Systems** tab in the main menu, then click on the **Add** button, right above the listed current systems. On the first page, just fill in the system name, and since you are connecting csv as a source file, you can set the system as **Read-only**. Then save the system settings - to make other options available.

Connector configuration

In the next step, switch to the **Configuration** tab of your new system. First, you need to choose a connector; in this case, it is the **CSV connector**. This will present you with the specific configuration for this connector.

Fill these fields:

- **Separator:** separator used in the csv file (comma "," in our example)
- **Header included:** does the file have a header (it usually does, as in our example; if not, it can be set in the "Header (multi)" field)
- **Source path:** path to the csv
- **Identifier:** the name of the column containing unique identifiers of the identities (LOGIN in our example)
- the rest is optional and should not be set in this case

Click Save.

Scheme

Next, go to the **Scheme** menu item of your selected system.

Have CzechIdM generate a scheme for you by clicking on the **Generate scheme** button. The result should look like this.

The screenshot shows a web interface for defining a scheme. At the top, there are two dropdown menus: 'System name' with the value 'Example csv system' and 'Object name' with the value '___ACCOUNT__'. Below these are 'Back' and 'Save and continue' buttons. The main section is titled 'Scheme attributes' and contains a table with the following data:

| | Name | Data type | Required | Multivalued | Id |
|--------------------------|---|------------------|-------------------------------------|--------------------------|---------|
| <input type="checkbox"/> | <input type="checkbox"/> FIRSTNAME | java.lang.String | <input type="checkbox"/> | <input type="checkbox"/> | c6b936d |
| <input type="checkbox"/> | <input type="checkbox"/> LASTNAME | java.lang.String | <input type="checkbox"/> | <input type="checkbox"/> | 0f20f80 |
| <input type="checkbox"/> | <input type="checkbox"/> __NAME__ | java.lang.String | <input checked="" type="checkbox"/> | <input type="checkbox"/> | e040a43 |

At the bottom right of the table area, it says '1 - 3 of 3 records'.

If your CSV file contains the header names in lower case and the scheme attributes weren't generated in upper case, go to each generated attribute and change its name to upper case.

Mapping

Now go to the **Mapping** menu item. Set how the file data is to be passed to CzechIdM.

First, set the following:

- **Operation type:** Synchronization
- **Object name:** __ACCOUNT__
- **Entity type:** Identity
- As **Mapping name** set whatever you want to, for example Synchronization of users.

Then map all columns as entity attributes the way you see it in the picture below. Set __NAME__ as an identifier.

Attribute mapping details

Disabled

Mapping name

sync users (Identity - Synchronization) ▼

Attribute in schema

FIRSTNAME (__ACCOUNT__) ✕ ▼

Name

FIRSTNAME *

User-defined name of the attribute

Strategy

Set value as it is ✕ ▼

Identifier

Entity attr.

Extended attr.

Main form definition is supported only.

Confidential attr.

Authentication attr.

Attribute used for authentication on connected system.

Include on password change

Send this attribute into provisioning, when password is changed.

Entity field

First name (String) ✕ ▼

IdM key

firstName *

Name of entity attr., name of extended attr., or key to confidential storage.

Synchronization

Finally, go to the **Synchronization** tab and add a new one, setting its **Name** (you can choose whichever name you like) and the fields as follows:

- **Allowed:** True
- **Reconciliation:** True (in this example, we don't use timestamps)
- **Set of mapped attributes:** Select the mapping from the previous step.
- **Correlation attribute:** `_NAME_`

For the rest of the configuration, you can keep the default values.

Afterwards, go back to the **Synchronization** menu and run the synchronization you've just created. You can check its result if you click on the **Logs** tab of your synchronization.

The screenshot shows the 'Synchronization details' page for an 'Example csv system'. The 'Logs' tab is selected, showing a table of synchronization logs. The table has columns for 'Running', 'Results', 'Started', 'Finished', and 'Id'. A single record is shown with the result 'Created entities' and ID '89e6431'. The 'Started' and 'Finished' times are both '08.12.2017 14:14:06' and '08.12.2017 14:14:07' respectively. The table indicates '1 - 1 of 1 records'.

Common issues

If the import does not go as expected, these are the things to check out:

- Possible database connection problems
- Permission of the folder with CSV files should be set to reading, writing and execution
- Owner of the CSV files and folder, including these files, has to be tomcat:tomcat
- `org.hsqldb.HsqlException: user lacks privilege or object not found: USER_EX65024` - The key column is not written in the configuration property **Column names (multi)** (there must be all columns)
- Check if the columns are lower or upper case - they should be upper case in the Scheme attributes, otherwise their values are not correctly loaded into IdM.

Your CSV is not valid:

- The columns in the header and in individual records must be the same
- No duplicates in identifier that you chose in the system configuration as unique
- No missing unique identifiers

- No empty columns in the header definition
- The CSV file has UTF-8 encoding and starts with BOM character (<U+FEFF>). It shouldn't start with BOM character.

From:

<https://wiki.czechidm.com/> - **IdStory Identity Manager**

Permanent link:

https://wiki.czechidm.com/tutorial/adm/how_to_identity_sync

Last update: **2021/03/03 12:52**

