Persons import

From the Person administration you can import persons into the system with a CSV file. In the **Administration panel > Persons** > click on the **Functions** button and select the option **Import**.

Please note that the import of CSV files is handled very restrictively. The CSV file must exactly match the specification, see specification-csv-for-person-import.

img

Besides selecting the correct csv file for the upload, you can now decide on 2 issues:

- what will happen to the persons who are already in the system and no longer contained in the csv file. These persons can be left as they are (no changes), they can all be deactivated, archived or even deleted
- you may decide to use the default password set for your system only for the users in the file that have no entry into the "password" column. (The standard password is selected individually and entered by a system administrator in the config file beforehand.)

img

When you select the appropriate radiobuttons and via "Select" chose the correct csv file, you can click on Continue.

Depending on the size of your file, the import can take a while, which you can see in the progress bar displayed during upload.

img

However, once the import is done, you will be shown a confirmation pop-up summarizing the import information that is saved in the database (name, source of error, ...)

img

> - **Archived persons:** Counts the persons whose status has been changed to "archived".
> - **Deactivated persons:** Counts the persons whose status has been changed to "deactivated".
> - **Activated persons:** Counts the persons whose status has been changed to "activated".
> - **Error:** Counts the produced errors through the import.
> - **New persons:** Counts the persons who have been newly created in the system by the import.

Specification CSV for Person Import

";" must be used for field separation.

Description	Config	Expected
Header	-	The easiest way to get a correct csv template is by first exporting a person. The csv file generated by this export can then be adjusted and used as an import csv.The header of a csv file must look as follows: img It is crucial that you do not change the column order. If you move a column to a different place, if you delete it or add a new column to the table, you cannot import any data anymore. Only data below the header (the 4 rows shown above) will be considered for the import. Should there be data above these rows, they will be ignored.
date	-	The date is editable and currently of no importance.
language	-	This is a mandatory field and must receive a value from the 6 available options: "de", "fr", "en", "it", "es", "zh". The entered language defines the language the system uses to double-check or import the properties (organisational units / activities).

Description	Config	Expected
encoding "ansi"	-	Defines the encoding (allows the unambiguous assignment of characters – must not be changed)
person-id	-	System ID of the user. Is automatically assigned by the system and must not be changed. Is used to identify a person when the related data is updated.
status	-	enabled archived disabled
name	-	Mandatory field
prename	-	Mandatory field
username	Optional identification	Mandatory field
password	-	**New person:** • if a password is set here, the user can login with that • if no password is set here (recommended way) the user can set himself a new password with the password forgotten flow during login **Updating existing persons:** • The password of an existing user remains untouched in the database, if the respective field is left empty on the import list. • Passwords are updated by entering a new password into the import list. The old password will be replaced by the new one.
email	Optional identification	Mandatory field. The system checks whether or not the entry is an e-mail address.
personal-id	Optional identification	- There is no defined minimal length - The maximal length is 255 characters - Not to be mistaken for "person-id" - Must not be unique

Description	Config	Expected
role	-	Following values are valid for this mandatory field: - learner - default-subadministrator - administrator Roles of a lower level cannot import or change roles of higher levels.
language	-	Mandatory field. Existing languages: de, en, fr, it, es, zh
orgunit, jobdescription**	-	- Organisation units / activities are not required (caution: for empty units / activities can not be searched). - If an organisation unit / activity is on the list, but not yet in the system, it will be newly created. - If a new organisation unit / activity is on the list and the system contains a different one, the existing organisation unit / activity will be removed and the new one added to the system. That is, the organisation unit / activity will be exchanged by the import. - If you wish to add a second organisation unit / activity to the system, the old organisation unit / activity must also be displayed on the list. To enter multiple organisation units / activities use the character " | " to separate them from each other. - If you wish to enter a path of an organisation unit / activity in the system, use the character " / " .
is_deletable	-	Defines whether the new person will be created with the setting "Can be deleted yes/no". There are the following settings for this mandatory field: 0 = No (person can not be deleted) 1 = Yes (person can be deleted)
change_password	-	Defines whether the new person will be created with the setting "Change password yes/no". There are the following settings for this mandatory field: 0 = No (person must not change her password at the next log-in) 1 = Yes (person must change her password at the next log-in) empty = no changes The field "Change_password" is empty in the export.

System configuration:

The **Systemadmin** configures both standard password generation and identification options under:

Default php in the "person_import" parameter.

> **Information about translations of organisation units/jobdescriptions (** in table)**
>
> *   An organisation unit / activity consists of an identifier for which there may be translations.
> *   If required, translations must be entered in the system in the properties administration.
> *   If there are no translations, the name of the identifier is used.
> *   Organisation units are displayed with path and translation in the selected export language. Translation changes can thus be properly updated with the import.
>   This is only done in the selected export language.

> **Edit existing person**
>
> The Person ID clearly identifies a person. If the ID is already included in the system, the existing person data is overwritten by the new data from the list. If the ID is not yet in the system or if the field is empty, a person can be identified with three more fields. Identification options are checked according to the following priorities:
>
> 1. "person_id" (defintion see "Specification CSV Content")
> 2. "personal_id" (defintion see "Specification CSV Content")
> 3. "email" (defintion see "Specification CSV Content")
> 4. "username" (defintion see "Specification CSV Content")

Process of error handling:

*   Validation CSV file and header
*   Content errors in the CSV
*   Validation export results
*   Error handling DB:
    *   Checks between the import list and the system occur on different levels. The database check displays an error but not its source on the GUI.
    *   There are two tables rendering information in the database. The table "trc_import_data" displays the source (line / column) of a faulty import.
    *   The two relevant tables are:
        *   trc_import_data
        *   trc_import_option

Error messages

```
if (error_msg === 'orgunits_not_accepted') {
        validatorData.headline = auxiliary.gt('Dateiüberprüfung fehlgeschlagen');

        validatorData.text = auxiliary.gt('Organisationseinheiten nicht akzeptiert!');

        validatorResult.push(validatorData);

        validatorData = [];

    }

    if (error_msg === 'too_many_header_lines') {

        validatorData.headline = auxiliary.gt('Dateiüberprüfung fehlgeschlagen');

        validatorData.text = auxiliary.gt('Falsche Anzahl Zeilen im Tabellen-Header!');

        validatorResult.push(validatorData);

        validatorData = [];

    }

    if (error_msg === 'header_fields_invalide') {

        validatorData.headline = auxiliary.gt('Dateiüberprüfung fehlgeschlagen');

        validatorData.text = auxiliary.gt('Ungültige Felder im Tabellen-Header!');

        validatorResult.push(validatorData);

        validatorData = [];

    }

    if (error_msg === 'no_person_header_found') {

        validatorData.headline = auxiliary.gt('Dateiüberprüfung fehlgeschlagen');

        validatorData.text = auxiliary.gt('Bitte überprüfen Sie den Tabellen-Header!');

        validatorResult.push(validatorData);

        validatorData = [];

    }

    if (error_msg === 'no_valide_person_found') {

        validatorData.headline = auxiliary.gt('Fehler');

        validatorData.text = auxiliary.gt('Keine gültigen Personen gefunden!');

        validatorResult.push(validatorData);

        validatorData = [];

    }

    if (error_msg === 'wrong_person_status') {

        validatorData.headline = auxiliary.gt('Fehler');

        validatorData.text = auxiliary.gt('Fehlerhafter Personenstatus!');

        validatorResult.push(validatorData);

        validatorData = [];

    }

    if (error_msg === 'wrong_person_username') {

        validatorData.headline = auxiliary.gt('Fehler');
```

```javascript
        validatorData.text = auxiliary.gt('Fehlerhafter Benutzername!');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
    if (error\_msg === 'wrong\_person\_role') {
        validatorData.headline = auxiliary.gt('Fehler');
        validatorData.text = auxiliary.gt('Fehlerhafte Benutzerrolle');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
    if (error\_msg === 'wrong\_person\_personal\_id') {
        validatorData.headline = auxiliary.gt('Fehler');
        validatorData.text = auxiliary.gt('Fehlerhafte Personalnummer');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
    if (error\_msg === 'wrong\_person\_language') {
        validatorData.headline = auxiliary.gt('Fehler');
        validatorData.text = auxiliary.gt('Fehlerhafte Benutzersprache');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
    if (error\_msg === 'wrong\_person\_is\_deletable') {
        validatorData.headline = auxiliary.gt('Fehler');
        validatorData.text = auxiliary.gt('Nicht erlaubte Einträge in "Ist löschbar"!');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
    if (error\_msg === 'duplicate\_person\_id') {
        validatorData.headline = auxiliary.gt('Fehler');
        validatorData.text = auxiliary.gt('Doppelte Personen-ID!');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
    if (error\_msg === 'empty\_username') {
        validatorData.headline = auxiliary.gt('Fehler');
        validatorData.text = auxiliary.gt('Benutzername ist leer!');
        validatorResult.push(validatorData);
        validatorData = \[\];
    }
```

```
if (error_msg === 'duplicate_username') {

    validatorData.headline = auxiliary.gt('Fehler');

    validatorData.text = auxiliary.gt('Doppelter Benutzername!');

    validatorResult.push(validatorData);

    validatorData = [];

}

if (error_msg === 'duplicate_personal_id') {

    validatorData.headline = auxiliary.gt('Fehler');

    validatorData.text = auxiliary.gt('Doppelte Personalnummer!');

    validatorResult.push(validatorData);

    validatorData = [];

}

if (error_msg === 'jobdescriptions_not_accepted') {

    validatorData.headline = auxiliary.gt('Fehler');

    validatorData.text = auxiliary.gt('Nicht erlaubte Einträge in "Tätigkeiten"!');

    validatorResult.push(validatorData);

    validatorData = [];

}

if (error_msg === 'orgunits_not_accepted') {

    validatorData.headline = auxiliary.gt('Fehler');

    validatorData.text = auxiliary.gt('Nicht erlaubte Einträge in "Organisationseinheiten"!');

    validatorResult.push(validatorData);

    validatorData = [];

}

if (error_msg === 'role_not_accepted') {

    validatorData.headline = auxiliary.gt('Fehler');

    validatorData.text = auxiliary.gt('Nicht erlaubte Einträge in "Benutzerrollen"!');

    validatorResult.push(validatorData);

    validatorData = [];

}
```