

# Systems - How to connect generic system

System connection configuration is initiated in the menu tab **Systems**. Above the list of current systems there is a button **Add**.

Systems

System name

System type

Connector server

CANCEL FILTER

FILTER

ADD

FILTER

	System name	Description	Asynchronous provisioning	State	Blocked operations	Id
<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> MS AD - Groups		<input type="checkbox"/>			ID: D0491651 TI: 9F911DD6
<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> MS AD - Users		<input type="checkbox"/>			ID: 659FE40F TI: E85359C5

1 - 2 of 2 records

## Basic information

[Click on it to connect new system.](#) On the new system page one must provide some basic information:

## New system

System name \*  
exampleSystem

Use remote connector server  
Use local connectors or select server ...

Local connectors will be used by default or select remote server with available connectors.

Password policy for validation  
Password policy for validation

☐ **Lower criticality of password policy for validation by role**  
Allow lowering the validation password policy criticality by overloading from the role

Password policy for generating  
Password policy for generating

☐ **Lower criticality of password policy for generating by role**  
Allow lowering the generating password policy criticality by overloading from the role

State  
System is active or choose other state ...

System state

☐ **Asynchronous provisioning**  
Active provisioning operations (create, update, delete), will run through the queue asynchronously. Change password operation will be still synchronous. Queue processing interval is configurable by long running task (ProvisioningQueueTaskExecutor).

☐ **Block create operation**  
All creation operations will be blocked

☐ **Block edit operation**  
All editing operations will be blocked

☐ **Block delete operation**  
All deletion operations will be blocked

Description

BACK SAVE AND CONTINUE

- **System name** - naming of your choice
- **Use remote connector server** - Connectors are means of interface between CzechIdM and other systems. Connectors run in a connector server. A local server provided by CzechIdM directly is usually used. So this checkbox will be usually unticked. There are some exceptions for specific connectors that must run remotely. For example connectors which call commands locally on the connected system server and therefore must be placed there. Exchange connector, for instance, uses calling of PowerShell commands directly on a domain controller server in an AD domain.
- **Password policy** for validation and generation - [see the chapter](#) about password policies.
- **Lower criticality of password policy for validation by role** - If this checkbox is selected, the password validation will use the criticality level defined for the role instead of the system policy.

- **Lower criticality of password policy for generating by role** - If this checkbox is selected, the password generation will use the criticality level defined for the role instead of the system policy.
- **Description** - an optional description of the system. It is customary to describe the purpose of the connected system, for example: "HR system - loading of job positions and departments".
- **State** - system states other than active:
  - **Readonly - with** provisioning queue - Systems marked in this way allow data reading only and are either source systems in CzechIdM or systems which are controlled but provisioning of data to them is intentionally prohibited for some time. In the latter case, all provisioned data is sent to the provisioning queue. The provisioning queue and history is displayed by: Systems → system detail (magnifying glass) → Provisioning. See the chapter [Audit](#).
  - **Readonly - without** provisioning queue - Systems marked in this way allow data reading only. Provisioning operations are not saved into queue, cannot be executed again. IdM account is created only (uid attribute only).
  - **Inactive - with** provisioning queue - Inactive systems do not allow even reading operations. If provisioning to such a system is to take place, then the operations end up in a queue as in the case of Readonly systems.
  - **Inactive - without** provisioning queue - Inactive systems do not allow even reading operations. Provisioning operations are not saved into queue, cannot be executed again. IdM account is created only (uid attribute only).
- **Asynchronous provisioning** - if the provisioning is asynchronous for the system, all the data is stored in the queue and managed by appropriate scheduled task. [Long running task](#) ProvisioningQueueTaskExecutor operates above the queue periodically and starts CREATED provisioning operation processing. Make sure you have **ProvisioningQueueTaskExecutor** configured, if you have some target system switched to use asynchronous provisioning. This is recommended option, since it significantly improves responsiveness of the application.
- **Block operations** - Block (create, edit or delete) operation will block checked operation.



Attribute values of Inactive systems with provisioning queue is available **are** calculated in the provisioning log.

## Configuration

Subsequently, a connector, which will connect the selected system, needs to be chosen in the tab **Configuration**. The configuration setting of the system connection always differs according to the selected connector.

When the connector is configured, the green button **Test Connector** can be used to test to

connection to real system.



Some connectors do not support "test" operation

The system of connectors provides connection to a system without the need to edit the administered system itself since their standardly provided interfaces are utilized.

The basic provided connectors are:

- Database Table Connector - connects a table in the database
- Scripted SQL Connector - connects any DB supporting JDBC.
- LDAP Connector - connects LDAP, even MS AD for some basic usage.
- CSVDirConnector - input/output from a CSV format text file

Other connectors can be added arbitrarily from publicly accessible [sources](#) (AD and Exchange connector), via own implementation or by inquiring of CzechIdM developers. Connectors use widely extended framework ConnId, formerly OpenICF, connector framework.

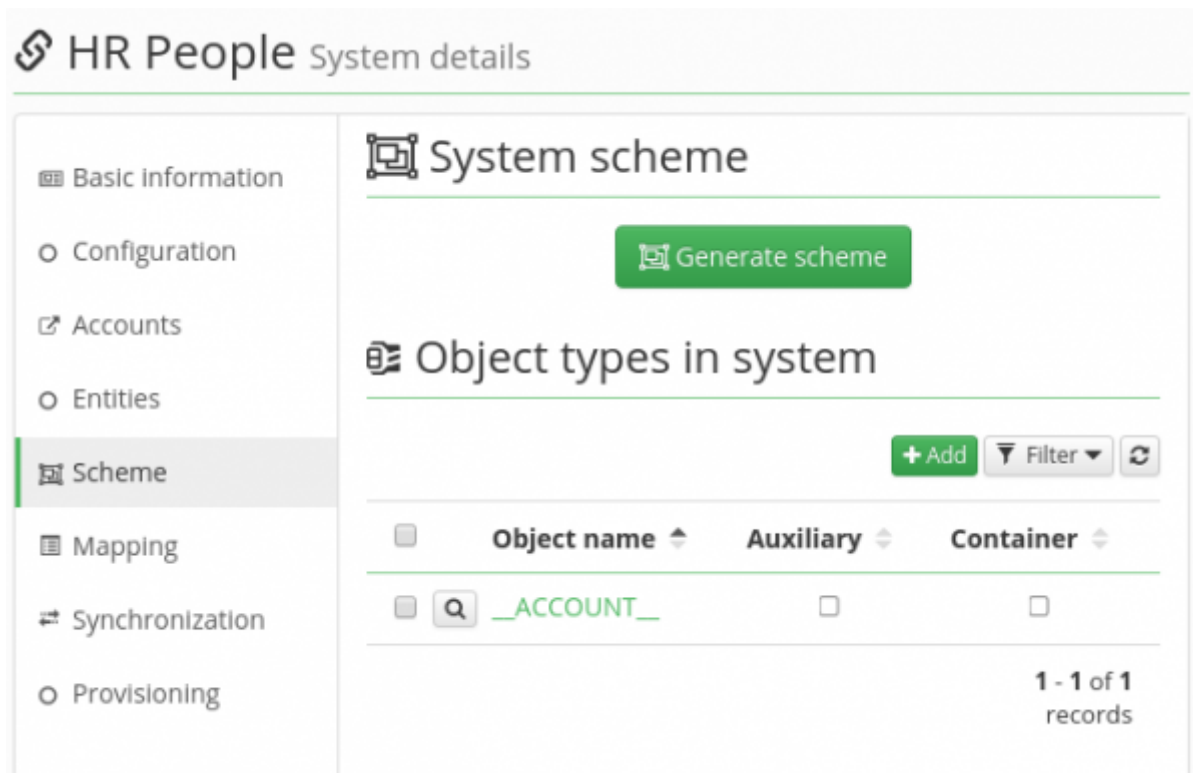


Additional connectors must be added to `_TOMCATHOME_/idm/WEB-INF/lib/` directory so IdM could load them properly. Moreover, if you don't see the newly added connector in the select box, clear the browser cache e.g. by closing and opening the browser window.

## Attributes Scheme

A scheme represents a list of attributes of some object (e.g. Account) in the connected system. By defining a scheme, CzechIdM is enabled to control management of object's attributes. The system scheme can be found in the tab **Systems → System detail → Scheme**.

The easiest and preferred way of how to create attributes scheme is to click the **Generate scheme**. Thus the attribute scheme is generated by the system's connector - all available attributes of the object are returned from the connector and can be modified by clicking on the object name in the table e.g. `__ACCOUNT__`.

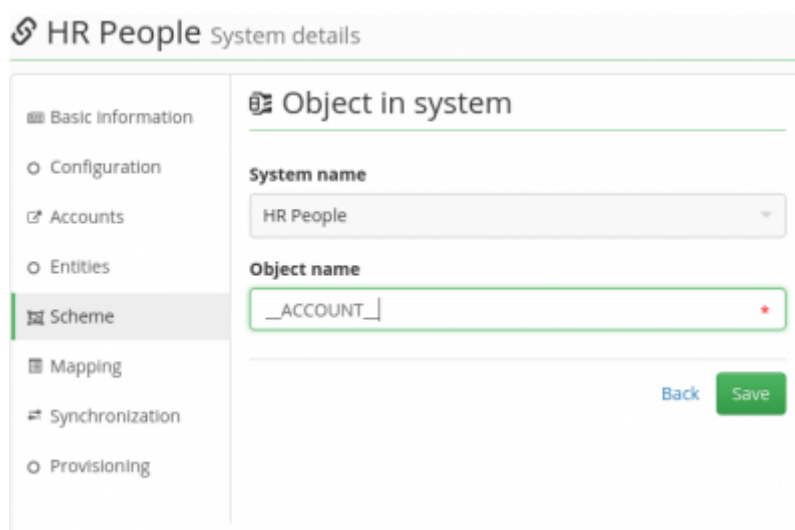


The screenshot shows the 'HR People System details' page. On the left is a sidebar with navigation links: Basic information, Configuration, Accounts, Entities, Scheme (selected), Mapping, Synchronization, and Provisioning. The main content area is titled 'System scheme' and contains a 'Generate scheme' button. Below this is a section titled 'Object types in system' with '+ Add', 'Filter', and refresh icons. A table lists object types with columns 'Object name', 'Auxiliary', and 'Container'. One record is shown: '\_ACCOUNT\_' with checkboxes for 'Auxiliary' and 'Container'. The bottom right of the table indicates '1 - 1 of 1 records'.



Not all connectors support automatic scheme generation. From the selection of standard connectors, this functionality is supported by Database Table connector and LDAP connector, for instance.

The other option of defining scheme is clicking on the green **Add button**, define the object e.g. `_ACCOUNT_` and then add attributes into the scheme manually one by one.



The screenshot shows the 'HR People System details' page with the 'Scheme' tab selected. The main content area is titled 'Object in system'. It contains two input fields: 'System name' with a dropdown menu showing 'HR People', and 'Object name' with a text input field containing '\_ACCOUNT\_'. Below these fields are 'Back' and 'Save' buttons.

If editing (magnifying glass by the attribute name), or creating (green Add button) attributes in scheme, their names on the system and their data types need to be filled in.

Object in system

System name

HR People

Object name

\_\_ACCOUNT\_\_

Back

Save

Scheme attributes

+ Add

Filter

	Name	Data type	Required	Multivalued
<input type="checkbox"/>	<input type="checkbox"/> EMAIL	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> GARANT	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> JMENO	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> __NAME__	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> POZICE	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> POZICE_ID	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> PRIJMENI	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> TITPRED	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> TITZA	java.lang.String	<input type="checkbox"/>	<input type="checkbox"/>

1 - 9 of 9 records

Usual data types are

- java.lang.String
- java.lang.Integer

All allowed types based on connid FrameworkUtil

- String.class
- Long.class
- Character.class
- Double.class
- Float.class
- Integer.class
- Boolean.class
- Byte.class
- byte[].class
- BigDecimal.class
- BigInteger.class
- Map.class

HR People

System details

Basic information

Configuration

Accounts

Entities

**Scheme**

Mapping

Synchronization

Provisioning

Attribute details

Attribute belongs to object

\_\_ACCOUNT\_\_

Name

EMAIL

Data type

java.lang.String

☐ Required

☒ able to read

☐ Multivalued

☒ able to create

☒ able to edit

☒ Returned by default

Back

Save



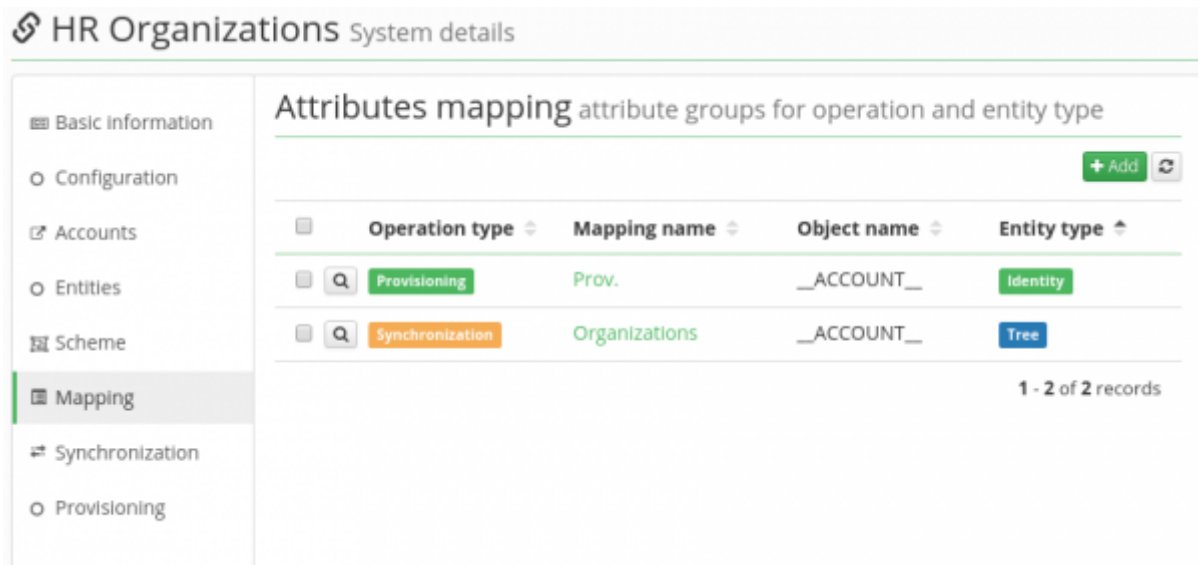
Every connector has some significant attributes, they are usually introduced by "\_\_" characters like \_\_NAME\_\_. Meaning of the attribute depends on the connector.

Then, some of the following settings can be enabled for each attribute:

- **Required** - attributes marked in this way are always sent to the end system (provisioning) regardless of whether the value in CzechIdM has changed compared to the value on the end system. In some cases, the connector specifically requires marking of some attributes as required. If it is not required, however, it is not recommended to use this option due to network load.
- **able to read** - It is recommended to leave this option allowed. Uncheck this option to ensure compatibility with connectors which do not allow reading some attributes.
- **multivalued** - CzechIdM allows loading and provisioning of attributes containing more values at the same time. For example, the attribute Titles can be set to be filled in from 2 attributes from CzechIdM – TitleBefore, TitleAfter. The attribute Titles must be then marked as multivalued in the scheme of the connected system.
- **able to create** - this option is used mainly when the connected system is both a source and end system. If your system is only source or only end, it is recommended to leave this option allowed. In this case reading and writing of the attribute can be controlled by the system configuration itself (ReadOnly or Inactive systems).
- **able to edit** - dtto
- **returned by default** - TODO Vítek

# Attributes mapping

When the attributes [scheme](#) for the object are ready, connected system attributes available in the scheme can be mapped onto CzechIdM entity attributes e.g. attributes of identity.



Attributes mapping is available at **Systems → Mapping**. If there is none use green Add button to create a new one.

HR People System details

Mapping of attributes for IdM entity and operation type

**Operation type**

Synchronization

**Mapping name**

HR People periodical synchronization

**Object name**

\_\_ACCOUNT\_\_

**Entity type**

Identity

Back Save

- **Operation type** - Firstly, the purpose of mapping needs to be selected. There are two options:
  - **synchronization** (upload data to CzechIdM)
  - **provisioning** (propagation of data from CzechIdM)
- **Mapping name** - Next the name for the mapping must be provided. The name is consequently displayed in synchronization and provisioning configuration
- **Object name** - sets which object type on the connected system will be mapped to CzechIdM



- **Entity type** - defines the entity in CzechIdM to which the object from connected system is mapped. Usual values are: Identity, Role, Role Catalogue, Contracted position, Tree

When the attribute mapping is created and it is clear what object in the connected system is mapped to what entity in CzechIdM, the procedure gets to the object/entity attributes configuration.

Click on the Add button to create a new attribute in current mapping.

**Attribute mapping details**

**BASIC SETTINGS** WHERE IS ATTRIBUTE USED?

☐ Disabled

Attribute in schema \*  
\_NAME\_ (ACCOUNT\_) x v

Name \*  
\_NAME\_

User-defined name of the attribute

☒ Identifier

☒ Entity attr.

☐ Extended attr.

Main form definition is supported only.

Entity field \*  
User name (String) x v

IdM key \*  
username

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

Strategy \*  
Set value as it is x v

☐ Send always

☐ Send IdM value only if its not null

☐ Confidential attr.

☐ Authentication attr.  
Attribute used for authentication on connected system.

☐ Include on password change  
Send this attribute into provisioning, when password is changed.

☐ Include only when criticality is changed to stronger  
The value of the attribute will only be sent to the target system when the aggregated password policy criticality is changed to stronger one. It will not be used in standard provisioning.

☐ Include only when criticality is changed to weaker  
The value of the attribute will only be sent to the target system when the aggregated password policy criticality is changed to weaker one. It will not be used in standard provisioning.

☐ Attribute with password  
Attribute will contain value of password. The attribute can't be override by role mapping. Into transformation will be add password in object GuardedString. Script must return null, or GuardedString.

☒ The value is cached  
The attribute value will be saved and read from the cache. At this moment, it is used only in sync. The key is this attribute and attributes from the end system (lcAttribute). The value is the transformed value from the end system.

These options can be filled:

- **Disabled** - If the attribute is disabled in mapping, it is not provisioned or synchronized.
- **Attribute in schema** - attributes from the connected system available in the current scheme.
- **Identifier** - Attribute is unique identifier of this object.
- **Entity attribute** - Attribute is part of entity.
- **Extended attribute** - Attribute isn't part of entity and his value and name is stored in EAV attributes.
- **Name** - Unique system identifier, this value is used in select boxes and in entities info
- **Strategy** - defines the strategy for the provisioning or synchronization. Available values:
  - Set value as it is - no standard transformation takes place
  - Merge - for multivalued attributes. Provision current idm attribute values + values returned from the connected system. This strategy is often used when connecting e.g. MS AD and CzechIdM manages placing the users into groups, but not all groups are loaded in CzechIdM itself.
  - Authoritative merge - preferred strategy to fill multivalued attributes. Only CzechIdM values are sent to connected system. If the attribute originally contained other values, they are replaced by the sent one.
  - Write only on create of the entity - If checked, the attribute value is sent to end system only together with CREATE operation. If UPDATE is sent to connected system, this attribute is not sent again. The same states for the synchronization.

- Write only if target value is null - If checked, only non existent attributes are filled in CzechIdM or connected system.



The attributes mapping must always contain one attribute marked as an **Identifier**. Otherwise the provisioning won't be possible.



Some of the options behaviour may vary depending on the connector used as well as connected system itself. e.g. some connectors returns NULL if the attribute does not exists, some connectors return empty string "" instead.

Other options of the mapped attribute are:


- **Send always** - Send this attribute to system always even if value isn't change (transformation rules is applied).
- **Send IdM value only if its not null** - Send this attribute only if value after transformation will not be null.
- **Confidential attribute** - Attribute value will be stored in confidential storage.
- **Authentication attribute** - With this attribute will be do authentication to end system (for example: username)
- **Include on password change** - Include this attribute when is provisioning password (reset, cahnge, create new)
- **Include only when criticality is changed to stronger** - The value of the attribute will only be sent to the target system when the aggregated password policy criticality is changed to stronger one. It will not be used in standard provisioning.
- **Include only when criticality is changed to weaker** - The value of the attribute will only be sent to the target system when the aggregated password policy criticality is changed to weaker one. It will not be used in standard provisioning.
- **Attribute with password** - Attribute will contain value of password. The attribute can't be override by role mapping. Into transformation will be add password in object GuardedString. Script must return null, or GuardedString.

**Entity field**User name (String) ✕ ▾**IdM key**username ★

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

**Transformation from system**


1 |

Insert script ▾ 

Allows value to be transformed from system into a form suitable for CzechIdM. Input parameters of this Groovy script are value of the attribute 'attributeValue' and list 'icAttributes' of object attributes in system.

**Transformation to system**

1 |

Insert script ▾ 

Allows value to be transformed from CzechIdM into a form suitable for connected system. Input parameters of this Groovy script are value of attribute 'attributeValue', IdM entity 'entity' and account identifier 'uid'. If output value is empty, system automatically uses available account identifier (uid).

[Back](#)[Save](#)

It is now clear what attribute is managed on the connected system and how the changes are propagated from/to the attribute. Obviously it is necessary to define what attribute in CzechIdM we want to connect the end system attribute to.

- **Entity field** - attributes from CzechIdM entity can be selected. This selection is available only if **Entity attr.** is enabled.
- **IdM key** - name of attribute in IdM, if administrator choose **Entity attribute** is this field read only and his value is set by entity field select box. If administrator choose **Extended attribute** is this field available for write and is necessary to enter name of extended attribute.

Now almost everything is set to synchronize or provision the attribute. If the range of standard options for attributes mapping is not wide enough, administrators can use [transformation scripts](#) to do advanced magic.

# Virtual Systems

This is a way of how to manage systems via user tasks, not directly via direct (e.g. network) communication. This feature is mainly implemented as [CzechIdM module](#).

From:

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

Permanent link:

<https://wiki.czechidm.com/tutorial/adm/systems>

Last update: **2024/08/12 10:06**

