

Provisioning

[identity](#), [role](#), [provisioning](#)

Just like synchronization, provisioning can be done for the following entities:

1. Identities (IdmIdentityDto)
2. Roles (IdmRoleDto)
3. Role catalogue items (IdmRoleCatalogueDto)
4. Tree nodes (structures) (IdmTreeNodeDto)

Provisioning of roles

Roles provisioning works differently than provisioning of identities. The main difference is the absence of a separate account management mechanism. Unlike identities where only those with a system provisioning role are passed, roles are propagated if and when they exist.

Provisioning of catalog

[role](#), [catalogue](#), [provisioning](#)

Role catalog provisioning also behaves differently than provisioning of identities. The main difference is the absence of a separate account management mechanism. Unlike identities where only those with a system provisioning role are passed, role catalogs are propagated if and when they exist.

In the case of role catalogue, account management is directly linked to the creation / modification / deletion event of the catalogue node.

Provisioning of tree nodes

[tree](#)

Tree provisioning behaves differently than provisioning of identities. The main difference is the absence of a separate account management mechanism. Unlike identities where only those with a system provisioning role are passed, tree nodes are propagated if and when they exist.

In this case, account management is directly linked to the creation / modification / deletion event of a tree node.

Retry mechanism

Provisioning operations ending with an error remain in the queue and new running time is scheduled to them = another attempt will be executed by long running task periodically – a long running task [RetryProvisioningTaskExecutor](#) configuration is needed. Only failed operations are processed from this queue by retry mechanism.

Asynchronous provisioning

The target system can be switched to use asynchronous provisioning - flag on the system detail. From then on, requests for active provisioning operations (create, update, delete) remain in the queue as ``CREATED`` and their processing is delayed. Operations in a queue are processed by a long running task [ProvisioningQueueTaskExecutor](#), which operates above the queue periodically and starts ``CREATED`` provisioning operation processing. Make sure you have the [ProvisioningQueueTaskExecutor](#) configured, if you want to switch any of the target systems to use asynchronous provisioning.



Change password operation is still synchronous – it is needed to change passwords immediately.

Provisioning of attachment

attachment

Since version **9.4.0** provisioning for **EAV attributes** with **attachment** is supported.

Example of use

- We have a system with an attribute **image-attribute**. That attribute has in the schema type array of bytes ([B]).
- This attribute is mapped on EAV attribute **image**. That means that there is also an **EAV attribute definition** for this attribute.



The EAV attribute must have the **Attachment** type set.

- If you run a provisioning session for an identity involving attachment for EAV attribute **image**, then the data (in byte array format) is propagated to the system (**image-attribute**).



No additional transformation is **required**. Load of attachment and transformation to the byte array is done automatically (**if transformation to the system is blank**).

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

Permanent link:
<https://wiki.czechidm.com/devel/documentation/adm/provisioning>

Last update: **2020/08/28 11:17**

