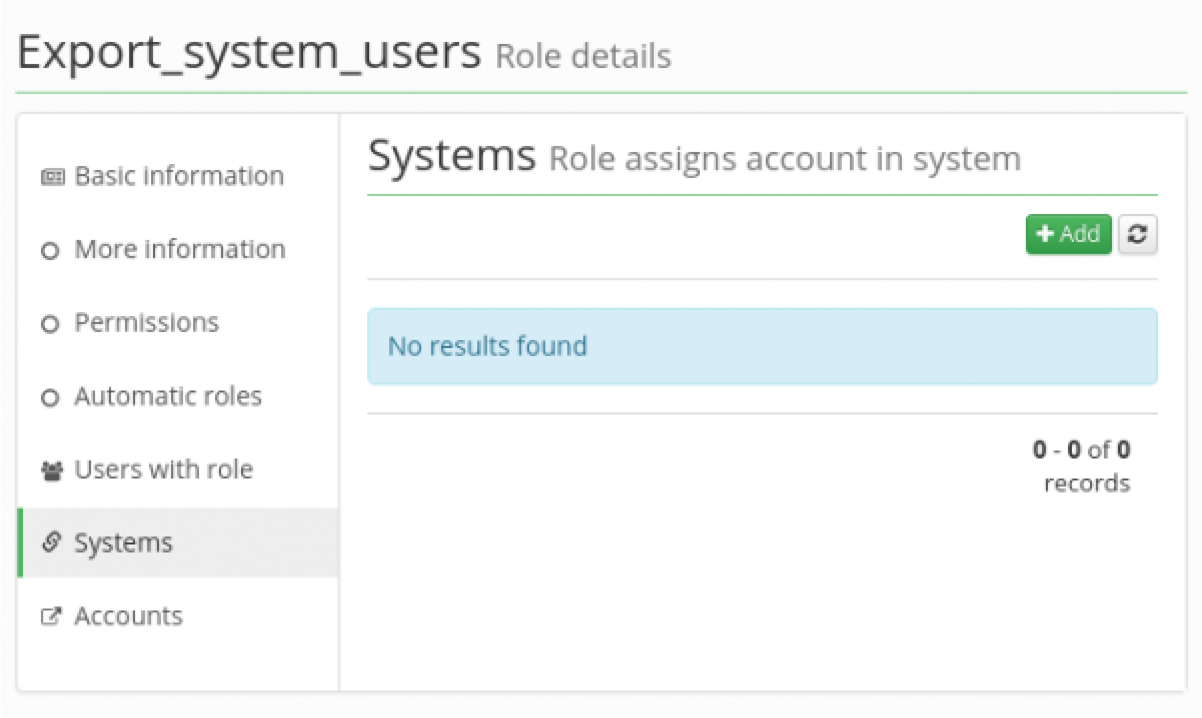
Provisioning - role and queue configuration

Provided there already exists an attributes mapping for provisioning we can define a role for identity provisioning and configure a provisioning queue.

Currently for Roles and Tree structures, all instances of an entity - i.e. all roles in CzechIdM - are provisioned to the system that has a provisioning mapping configured.

Configuring a role for provisioning

Provisioning for an identity can be executed only if a user has assigned a role which assigns accounts in the system and uses some of the system attribute mapping. To configure a role, go to the **Role** tab and choose the role that will assign the account in the system, e.g. the role "LDAP" or create a new one. Click on the role detail (magnifying glass next to the role name) and switch to the **Systems** tab.

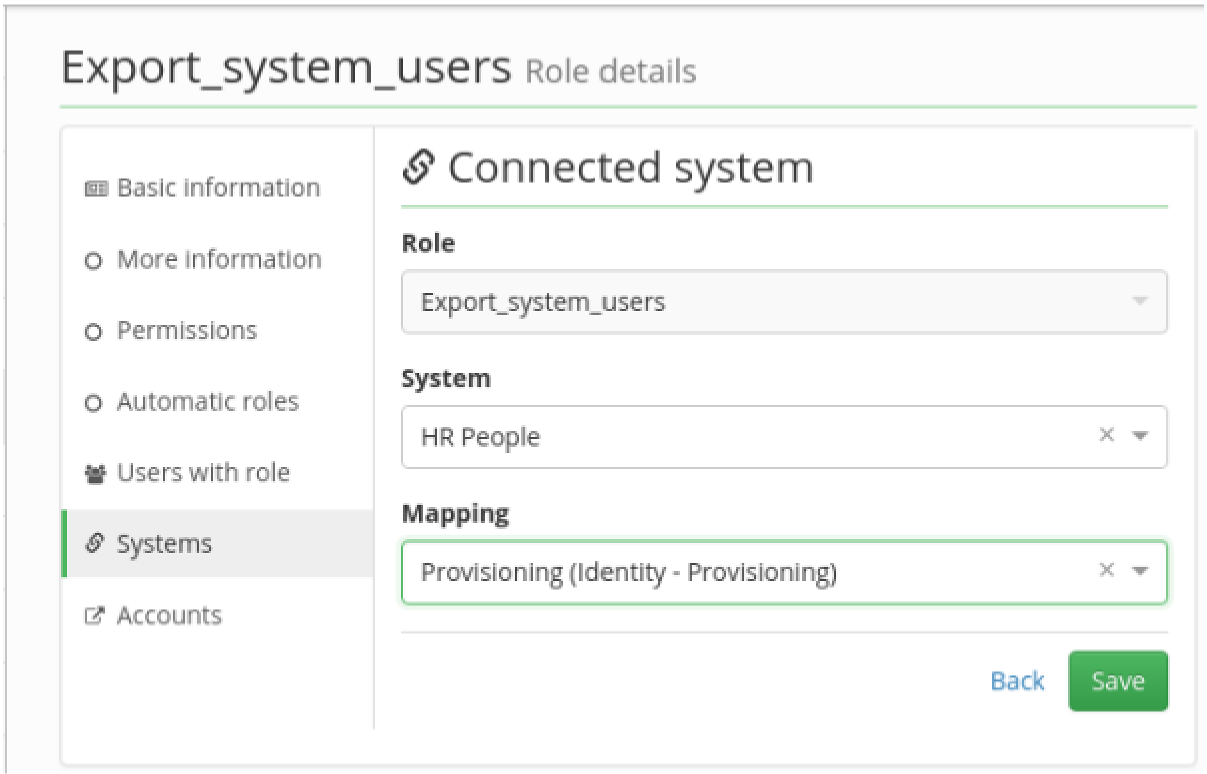

Export_system_users Role details

Systems Role assigns account in system

No results found

0 - 0 of 0 records

This tab contains a list of provisioning mappings. The list is usually empty or contains one item. Roles usually don't assign multiple systems to avoid complexity, but it can be done. For adding new system's mapping click on the **Add** button.



Export_system_users Role details

Connected system

Role: Export_system_users

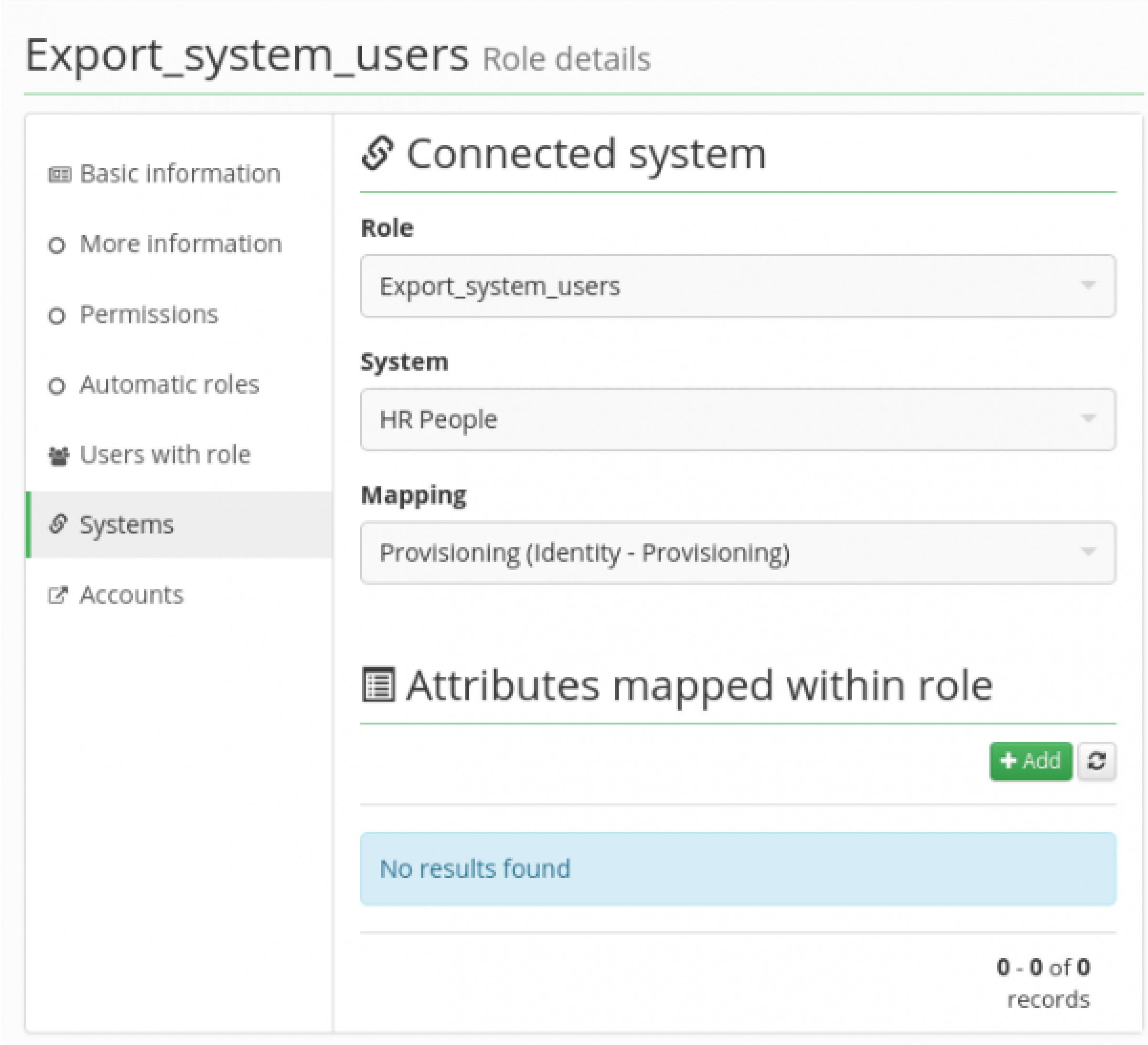
System: HR People

Mapping: Provisioning (Identity - Provisioning)

Set up the following fields:

- Role – ReadOnly field containing the name of the processed role
- System – The name of the managed system that will be assigned by the role to make provisioning. E.g. "LDAP".
- Mapping – The provisioning mapping which was defined earlier for the system.

After the configuration is saved - **Save button** - new options section appears:



Export_system_users Role details

Connected system

Role

Export_system_users

System

HR People

Mapping

Provisioning (Identity - Provisioning)

Attributes mapped within role

No results found

0 - 0 of 0 records

The next section "**Attributes mapped within a role**" can be used to override the selected mapping from the above. Overriding an attribute is used for setting a value which is different from the value in the system mapping. For example, if the role should assign a specific permission on the system, this specific permission can't be in any way set in the general attribute mapping for the system. E.g. this would be used for the attribute __MEMBER_OF__ of the MS Active Directory and roles that assign specific AD groups.

When a user has at least one role, which contains this configuration of attribute mapping for the provisioning, the user is automatically provisioned to the given system. The provisioning takes place after every change of some of the mapped attributes.

Provisioning queue

Propagation of data in CzechIdM is performed by the queue containing requests for the managed systems. When some attribute of an identity changes and the identity has account on some managed system, a new operation CREATE/UPDATE/DELETE is written to the provisioning queue as an **active operation**. The same principle holds for all objects that support provisioning:

- Identity
- Role

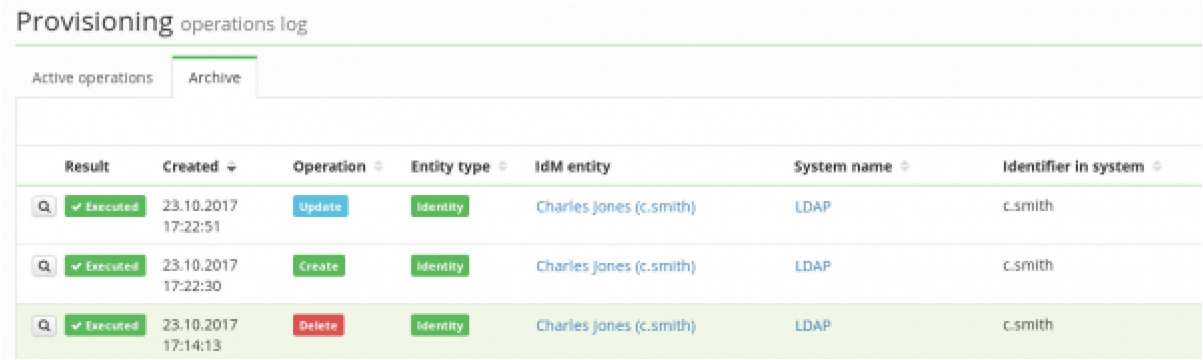
- Organization (TreeNode)
- Role catalogue

The audit of provisioning can be accessed in the following manner:

- **Audit → Provisioning**
- **Systems → Detail (magnifying glass) → Provisioning** – this way is actually only a shortcut to the previous location, the shortcut which automatically fills the filter "System" by the value of the selected system. Note that when you leave this form, switching to another tab in the menu, to return back later through the Audit → Provisioning, the filter stays filled with the original values of the system. As a result, you won't see the whole provisioning audit, only the filtered parts. To cancel the filter, click on the **Filter** button and then **Cancel filter**.

Either way you get to a form containing two tabs:

- Active operations – the tab displays the queue of pending requests for provisioning
- Archive – the tab displays already processed requests for provisioning



Provisioning operations log

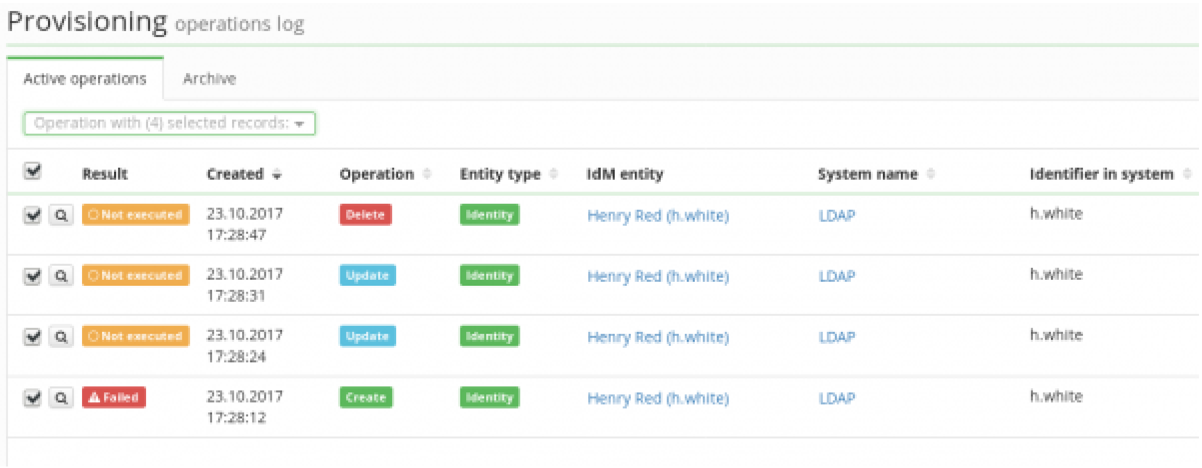
	Result	Created	Operation	Entity type	IdM entity	System name	Identifier in system
🔍	✓ Executed	23.10.2017 17:22:51	Update	Identity	Charles Jones (c.smith)	LDAP	c.smith
🔍	✓ Executed	23.10.2017 17:22:30	Create	Identity	Charles Jones (c.smith)	LDAP	c.smith
🔍	✓ Executed	23.10.2017 17:14:13	Delete	Identity	Charles Jones (c.smith)	LDAP	c.smith

The logic behind the provisioning queue is as follows: If there is a waiting operation for an object, say the user "j.doe", then any other operation for this object is added to the queue. So we can have a queue containing a list of pending operations for the user "j.doe": CREATE, UPDATE, UPDATE, DELETE.

Manual retry of provisioning operations

The most common reasons a provisioning operation does not run include these: unavailability of the system, wrong configuration of the system's connection, setting the system read-only. The operation will then result in an error and it will wait in the queue for further processing. We can repeat the action manually in CzechIdM - select this operation and choose the desired action from the select box **Operation with selected record**. There are two available actions:

- Retry operation
- Cancel operation



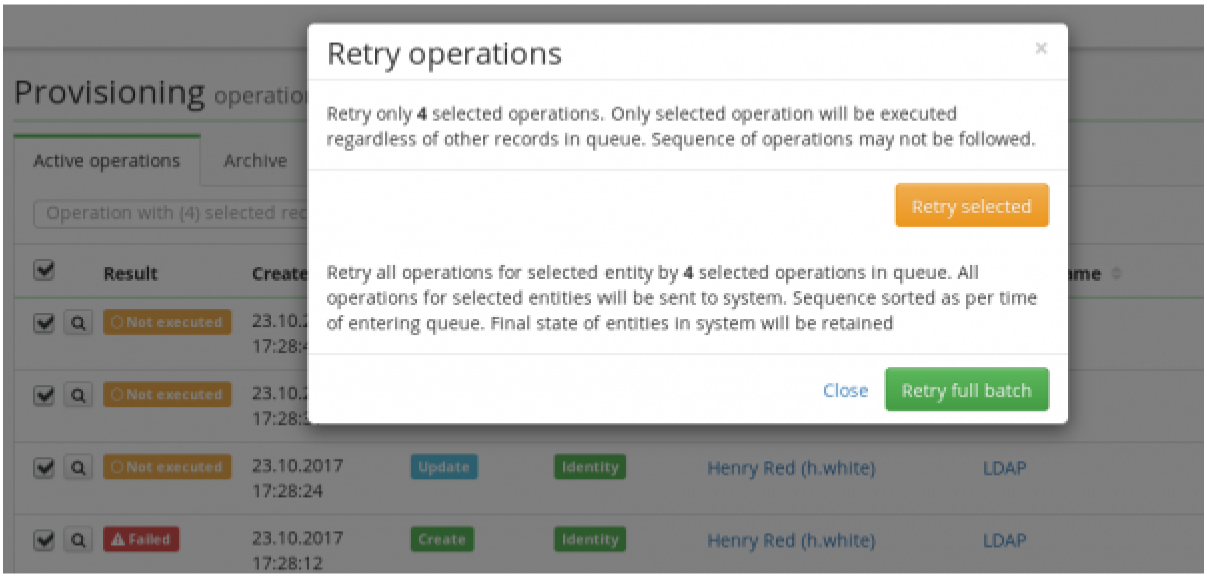
Provisioning operations log

Active operations | Archive

Operation with (4) selected records: ▾

☑	Result	Created	Operation	Entity type	IdM entity	System name	Identifier in system
☑	Not executed	23.10.2017 17:28:47	Delete	Identity	Henry Red (h.white)	LDAP	h.white
☑	Not executed	23.10.2017 17:28:31	Update	Identity	Henry Red (h.white)	LDAP	h.white
☑	Not executed	23.10.2017 17:28:24	Update	Identity	Henry Red (h.white)	LDAP	h.white
☑	Failed	23.10.2017 17:28:12	Create	Identity	Henry Red (h.white)	LDAP	h.white

In both cases, we will be asked just before the actual processing, if we want to retry/cancel **the full batch**, or only the **selected** records.



Retry operations

Retry only **4** selected operations. Only selected operation will be executed regardless of other records in queue. Sequence of operations may not be followed.

Retry selected

Retry all operations for selected entity by **4** selected operations in queue. All operations for selected entities will be sent to system. Sequence sorted as per time of entering queue. Final state of entities in system will be retained

Close | Retry full batch

If we choose retrying the full batch, all operations for this specific object in the queue (not only the selected operations!) are retried/cancelled in the same order as they are organized in the queue. E.g. when you select CREATE operation on the user "j.doe" and choose "Retry full batch", then all operations for the user "j.doe" will be retried in the respective order CREATE - UPDATE - UPDATE - DELETE.

If we choose the option **Retry selected**, the selected operations are retried/cancelled in the same order as they are organized in the queue. All the other not selected operations stay waiting in the queue. E.g. when you select CREATE and UPDATE for the user "j.doe" and choose "Retry selected", the CREATE and then the UPDATE operation will be retried and second UPDATE and DELETE will remain in the queue without any change.

Retrying selected operations requires knowledge about connecting the system. If an administrator chose some operations that don't make sense - e.g. choosing DELETE without choosing CREATE first - the processing would result in an error.

Automatic retry of provisioning operations

The long-running task (LRT) named **RetryProvisioningTaskExecutor** exists in CzechIdM for the sake of automatic retry of provisioning operations. This LRT executes the provisioning queue in defined intervals. If you want CzechIdM to periodically retry failed provisioning operations from the queue, enable and set this task.

If there is a planned unavailability of a managed system, we recommend that you turn off this task temporarily for performance reasons.

Detail of a provisioning operation

The detail of a provisioning operation can be accessed by clicking on the magnifying glass next to the record in the provisioning queue. There are the following fields:

- Created - the date of requesting the operation
- Operation - the type of operation (Create/Update/Delete)
- Entity type - e.g. Identity, TreeNode, …
- IdM entity - the name of the provisioned entity in CzechIdM
- System name - the system which is provisioned into
- Identifier in system - the name of the provisioned entity in the managed system
- Result - in the case of an error, there is additional information about the error, such as:
    - error code - e.g. `PROVISIONING_ATTRIBUTE_VALUE_WRONG_TYPE`
    - error description - e.g. "Provisioning - attribute value is not in correct type"
    - stacktrace - Java stacktrace of the exception, e.g. `eu.bcvsolutions.idm.acc.exception.ProvisioningException: Schema attribute __ENABLE__ defines type java.lang.String. But value type is java.lang.Boolean!`

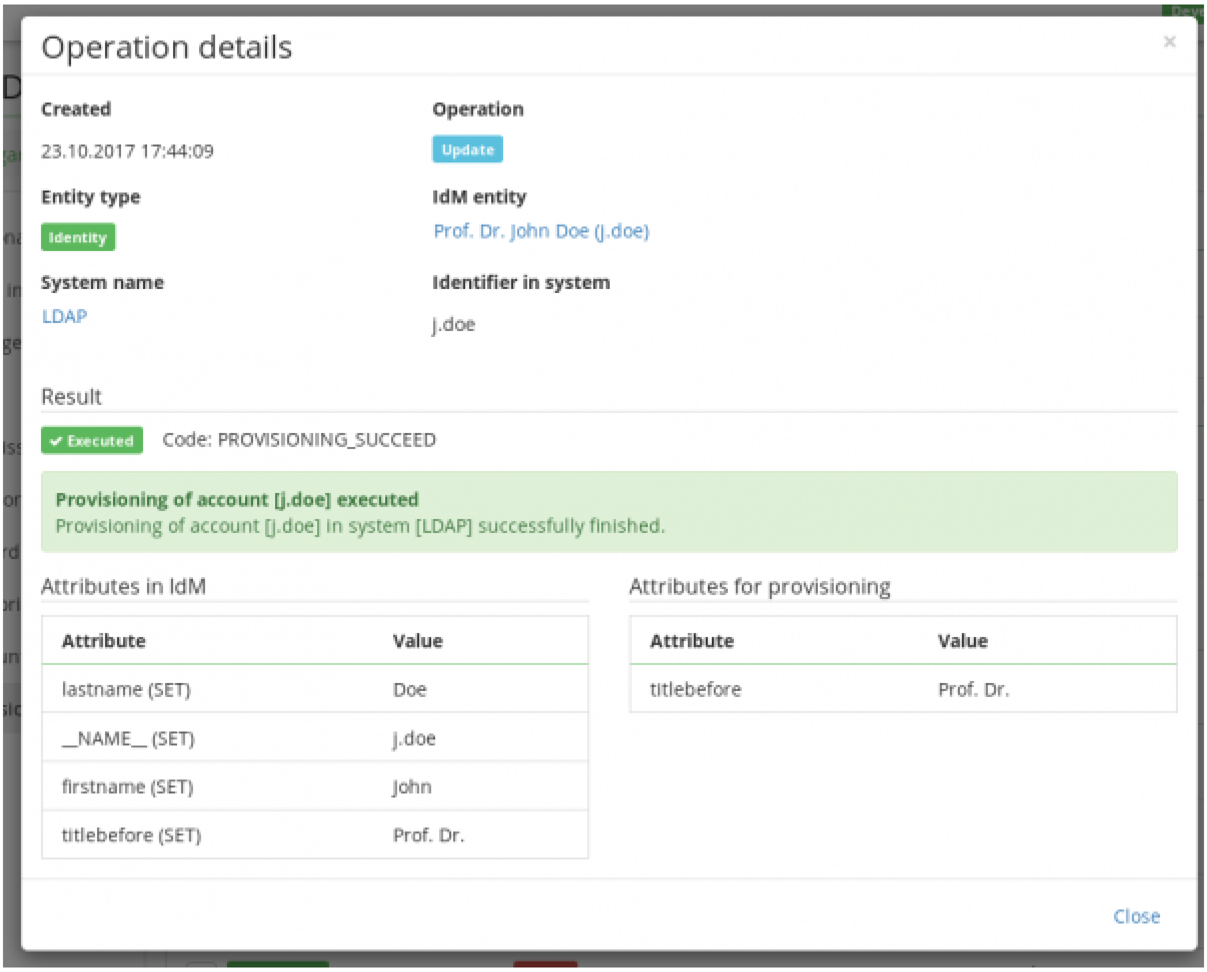
This information can be used to determine the reason for the error. In the example above, the reason is evidently the wrong configuration of a provisioning schema; namely, it's necessary to set the attribute __ENABLE__ to acquire Boolean, or use a transformation script to transform Boolean to String.

The detail of the operation displays two tables:

- **Attributes in IdM** - the left table displays all attributes that are set in the system mapping for the given entity. E.g. the system "LDAP" maps the attributes *name, firstname, lastname, titlebefore* for the entity "Identity". The table contains these attributes and their corresponding values, e.g. *j.doe, John, Doe, Dr.*. We call this table as the **wish**, i.e. the desired values to send. The value of every attribute is final, meaning that if their mapping required a transformation or a computation, the transformation had been already applied.

E.g. if the mapping contains the attribute *Manager*, its value could be "*CN=John Doe, O = My Company*", which doesn't need to be saved anywhere in IdM, because it is computed by the transformation script during provisioning.

- **Attributes for provisioning** – the right table displays selected attributes and their values, which are really propagated to the managed system. Not all attributes have to be sent to the system. Namely the attributes whose actual value on the system **doesn't differ** from the wish (the left table). The exception to this rule is, of course, the attributes that were marked **required** in the provisioning mapping.



Operation details

Created

23.10.2017 17:44:09

Operation

Update

Entity type

Identity

IdM entity

Prof. Dr. John Doe (j.doe)

System name

LDAP

Identifier in system

j.doe

Result

✓ Executed Code: PROVISIONING_SUCCEED

Provisioning of account [j.doe] executed
Provisioning of account [j.doe] in system [LDAP] successfully finished.

Attributes in IdM

Attribute	Value
lastname (SET)	Doe
__NAME__ (SET)	j.doe
firstname (SET)	John
titlebefore (SET)	Prof. Dr.

Attributes for provisioning

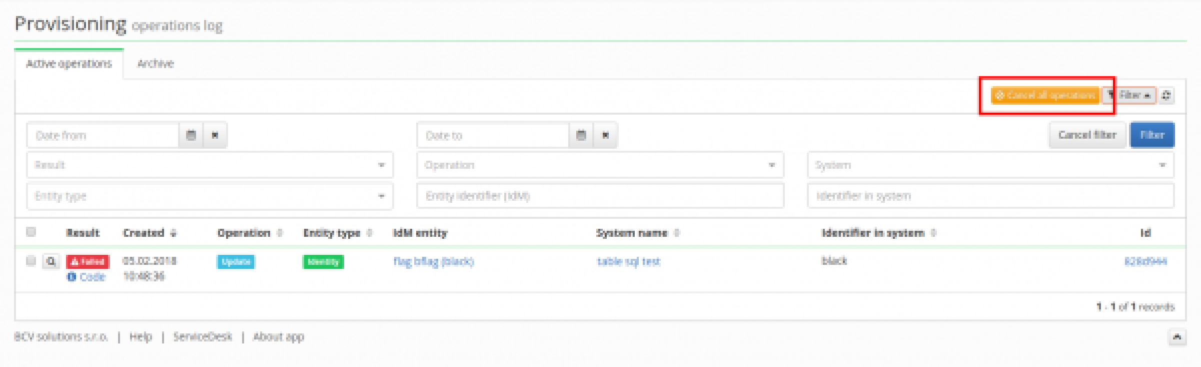
Attribute	Value
titlebefore	Prof. Dr.

The right-hand table may be empty, namely in these two cases:

- The provisioning queue already contains an older operation for the same entity. In such case, IdM doesn't compute the difference between the wished and the actual state of the system account for performance reasons. That would require unnecessary network communication. Also, the actual state of the system account may have been changed during the period of waiting of the operation in the queue. E.g. the administrator may have changed the value in the end system. The information in the queue could be misleading in such case.
- There is no difference between the attributes that would be sent (the wish) and the values in the managed system. This could happen only if no attribute in the provisioning mapping is marked as required.

Cancel provisioning queue

A new feature is available that will come in handy in configuring provisioning. When a provisioning queue has many operations and even cancel operation (**Operation with selected record**) is not very efficient, there is now the **Cancel all operations** button. This button uses a provisioning queue filter and cancels the batches of all found operations. So even if an operation is not found with this filter, but another operation in the same batch is found, both of them are cancelled. Cancelled operations are moved to the archive. This feature is only accessible to a user with admin authorization.



Result	Created	Operation	Entity type	IdM entity	System name	Identifier in system	Id
Failed Code	05.02.2018 10:48:36	Update	Identity	flag bflag (black)	table sql test	black	828d944

1 - 1 of 1 records

Provisioning brake

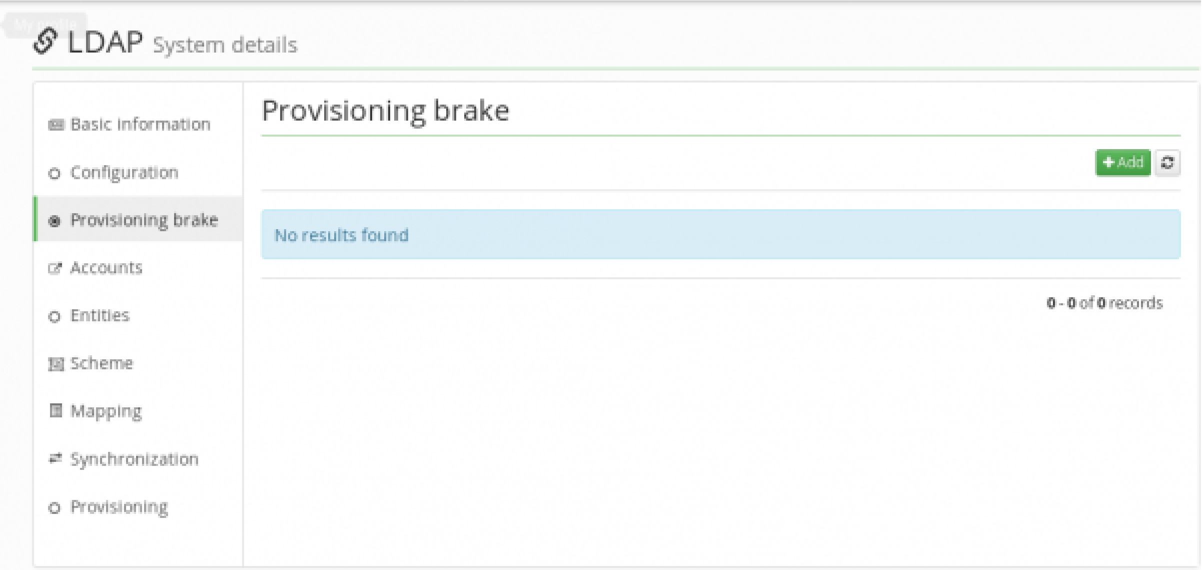
The provisioning brake is a tool to monitor how many times the specific provisioning operation (create, update, delete) is done. It is also possible to set **warning** or **disable** limit for each operation, which is very useful for business critical systems. After the limit is exceeded (either warning or disable), a notification will be sent to all recipients for specific provisioning break configuration. After the **disable** limit is exceeded for the operation, **the operation won't be executed any more**, until administrators manually check the current situation.

It's also possible to configure a global provisioning brake, which will be applied to all systems without the need to configure the brake separately for every managed system.

Configuring the provisioning brake

In the following example, we want CzechIdM to monitor all Delete operations on the managed system. We don't expect that the accounts on the system would be deleted too often. On the contrary, we want to be notified whenever CzechIdM deletes more than 2 accounts in the period of 60 minutes, because the system is business critical. If CzechIdM deletes more than 5 accounts in one hour, it's probably some kind of error in HR system or in the configuration. In such case, we want CzechIdM to cancel all following delete operations and we will resolve the situation manually.

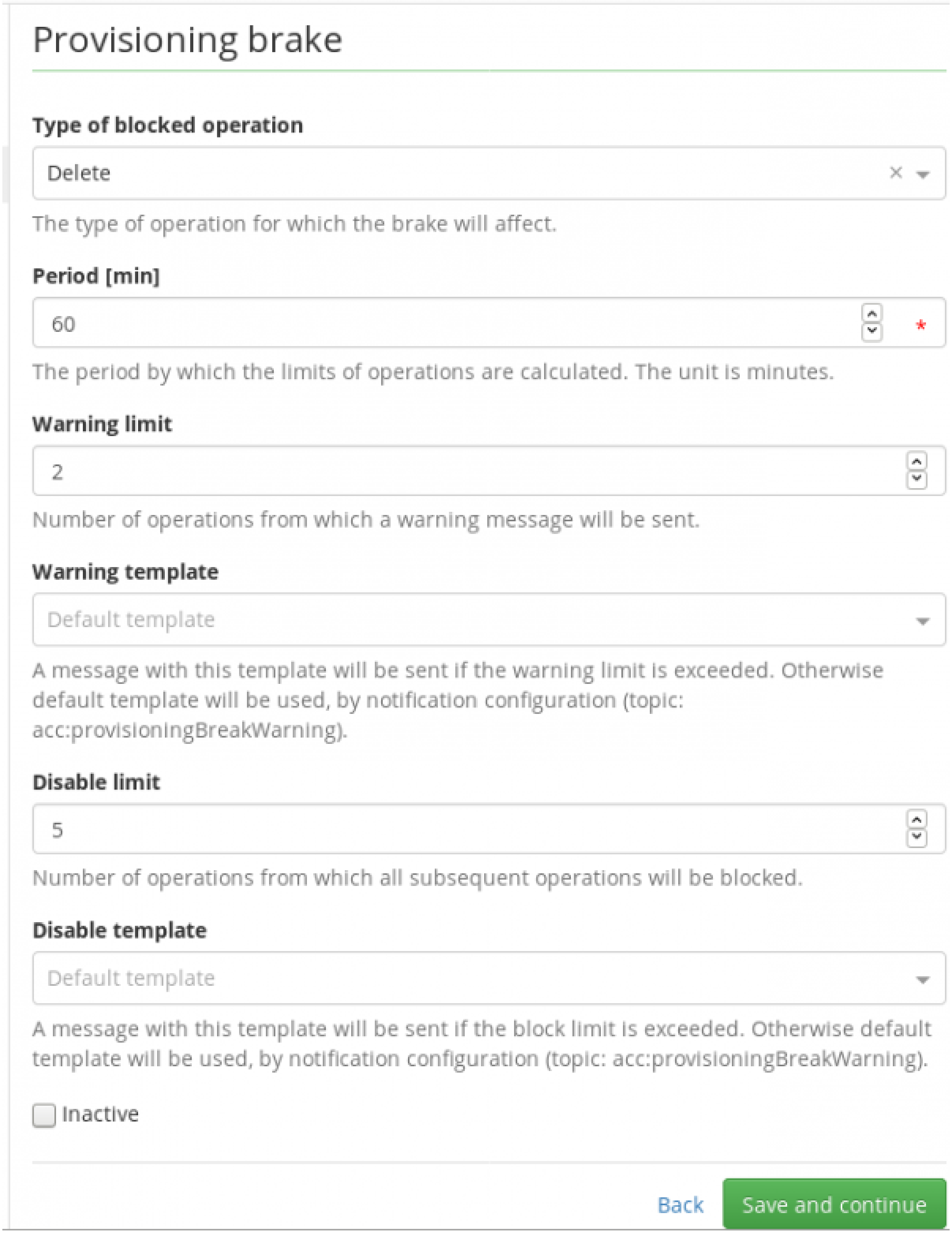
The provisioning brake for the specific system can be configured in **Systems → Detail (magnifying glass) → Provisioning brake**. To add a new configuration, click on the button **Add**.



No results found

0 - 0 of 0 records

Select the type of operation, which will be monitored by the provisioning brake. The possible values are Create, Update, or Delete. In our example, choose Delete. Then set the period for evaluating limits (in our example 60 minutes), the warning limit for the number of operations after which the warning notification is sent (in our example 2), and the disable limit - the maximum number of operations processed by CzechIdM in the configured period (in our example 5). Then click on the button **Save and continue**.



Provisioning brake

Type of blocked operation

Delete

The type of operation for which the brake will affect.

Period [min]

60 *

The period by which the limits of operations are calculated. The unit is minutes.

Warning limit

2

Number of operations from which a warning message will be sent.

Warning template

Default template

A message with this template will be sent if the warning limit is exceeded. Otherwise default template will be used, by notification configuration (topic: acc:provisioningBreakWarning).

Disable limit

5

Number of operations from which all subsequent operations will be blocked.

Disable template

Default template

A message with this template will be sent if the block limit is exceeded. Otherwise default template will be used, by notification configuration (topic: acc:provisioningBreakWarning).

☐ Inactive

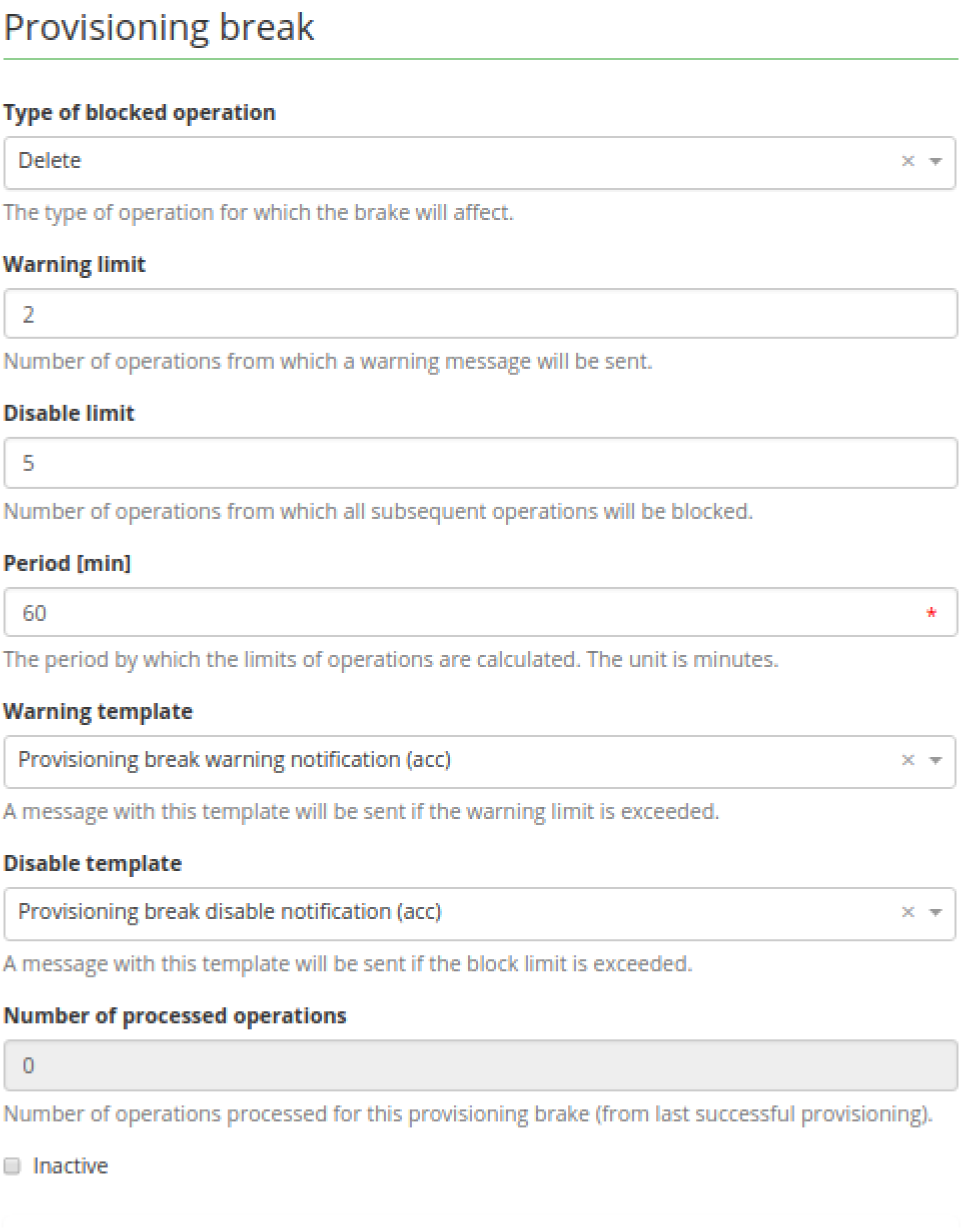
Provisioning brake configuration attributes

All possible settings of the provisioning brake:

Attribute	Description
Type of blocked operation	Type of blocked operation is one of **required attributes** and determines for which operation (create, update or delete) it is intended.

Attribute	Description
Warning limit	After the system exceeds this limit, the notification with warning about this limit is sent. This notification will be sent only once after exceeding the limit. Next attempts after exceeding the limit send no notification.
Disable limit	After the system exceeds this limit, the notification with information about disabling the system for this operation is sent. Moreover, the system will be disabled for this specific operation.
Period [min]	Period in minutes, a required attribute. The limits are evaluated for number of operations per this period.
Warning template	Specific template for warning. Notification configuration (topic) has also defined default notification template.
Disable template	Specific template for disable. Notification configuration (topic) has also defined default notification template.
Number of processed operations	This is the actual count of processed operations. (This field is displayed only when editing an existing configuration.)
Inactive	If the brake is Inactive, its settings doesn't affect provisioning at all. (This field is displayed only when editing an existing brake.)

Example configuration for delete operation:



Provisioning break

Type of blocked operation

Delete

The type of operation for which the brake will affect.

Warning limit

2

Number of operations from which a warning message will be sent.

Disable limit

5

Number of operations from which all subsequent operations will be blocked.

Period [min]

60 *

The period by which the limits of operations are calculated. The unit is minutes.

Warning template

Provisioning break warning notification (acc)

A message with this template will be sent if the warning limit is exceeded.

Disable template

Provisioning break disable notification (acc)

A message with this template will be sent if the block limit is exceeded.

Number of processed operations

0

Number of operations processed for this provisioning brake (from last successful provisioning).

□ Inactive

The example configuration means: If the system processed more than 2 operations during last 60 minutes, the warning notification would be sent. If more than 5 delete operations are processed, the system would be immediately blocked for delete operations (so the 6th operation wouldn't be processed) and also the notification with information about disabling the system would be sent. Both this limits are evaluated for 60 minutes.

The disable limit contains the maximum number of successfully processed operations. If the value is 5, CzechIdM enables to process 5 operations. However, requesting the 6th operation will be blocked by the system.

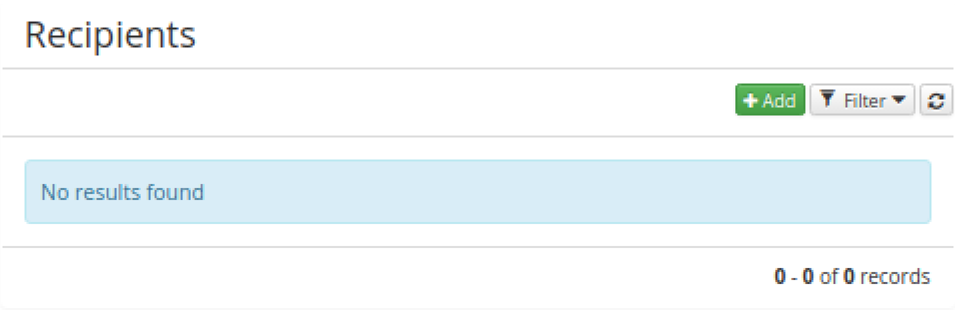
Each system can have a maximum of one provisioning brake configuration for a single provisioning event type (create/update/delete).

Recipients

After saving the new configuration, add the recipients which will receive the notifications. **It's not possible to add** recipients before you create the provisioning brake configuration. Of course, it's possible to edit the recipients for an existing provisioning brake.

The recipient can be an identity or a role. If you choose a role as the recipient, the notification will be sent to all identities that have this role assigned.

The recipient is added from the Recipients table (button + Add)



Recipients

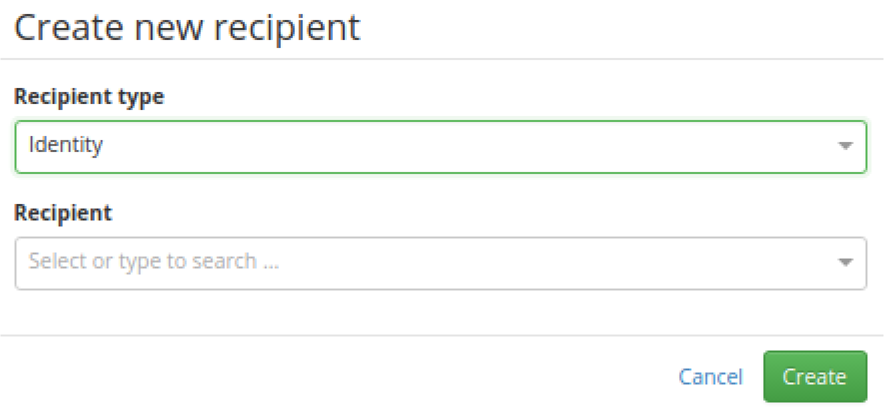
+ Add | Filter | ⟳

No results found

0 - 0 of 0 records

For both types (identity or role), the detail of recipient looks similar, see pictures:

Identity recipient detail:



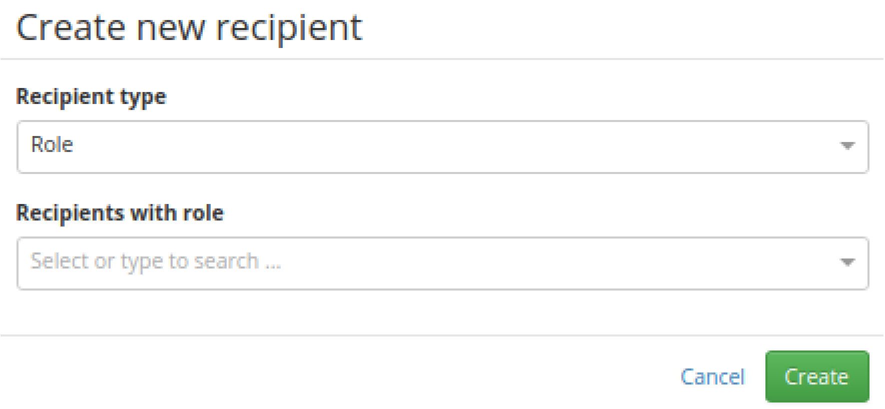
Create new recipient

Recipient type: Identity

Recipient: Select or type to search ...

Cancel | Create

Role recipient detail:



Create new recipient

Recipient type: Role

Recipients with role: Select or type to search ...

Cancel | Create

Data integrity is checked before you delete identity or role that is set as the provisioning brake recipient.

Blocked operations

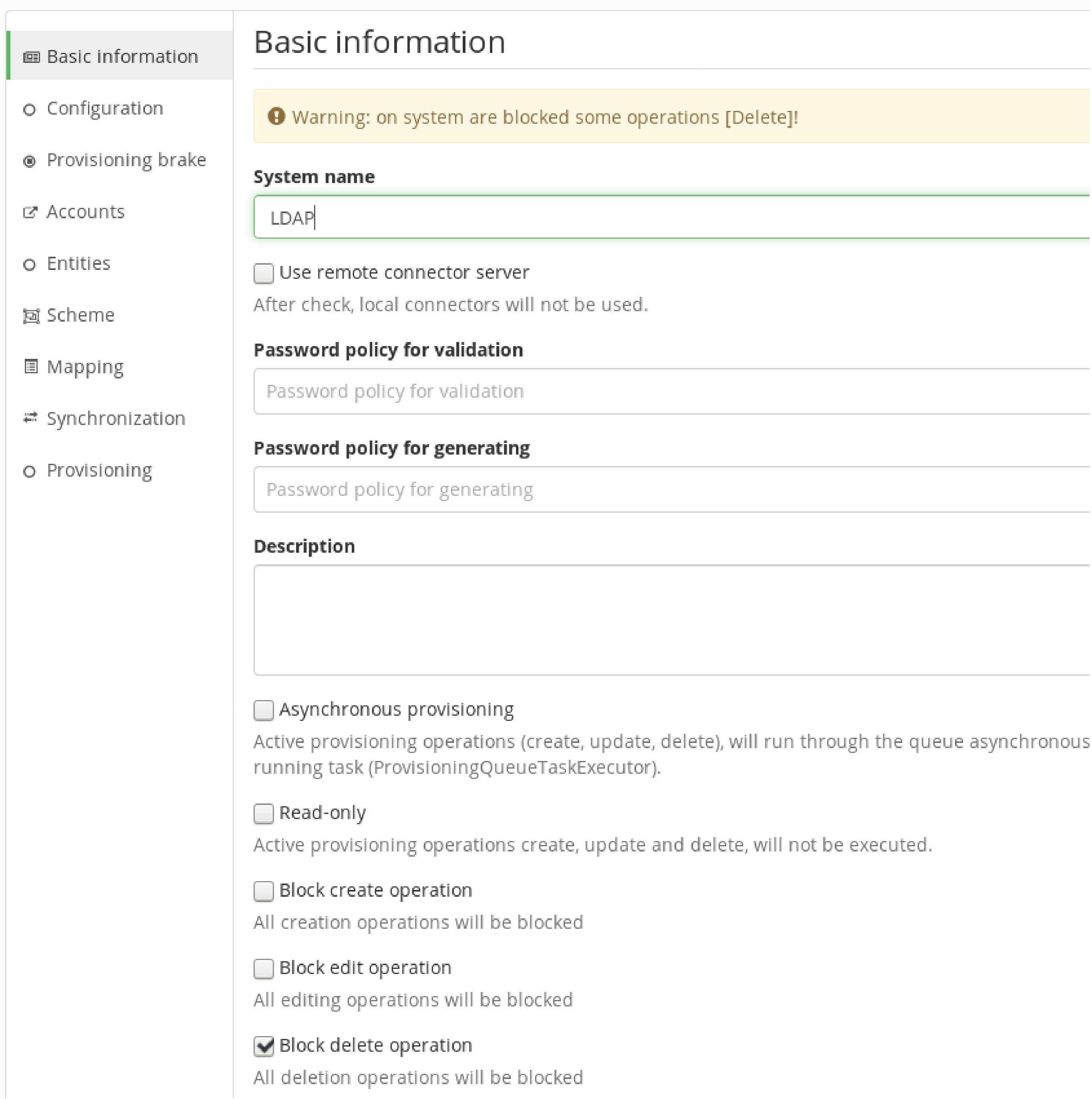
When the disable limit of the provisioning brake configuration is exceeded, the system is marked by one of these flags: **Block create operation**, **Block update operation**, **Block delete operation**. The flag can be also set manually in the system's configuration, the result is the same as when the provisioning brake takes effect: the corresponding operation (create, update or delete) is blocked.

If you want to enable the operation again, set the corresponding flag from true to false and save the system's configuration. This will also reset the provisioning brake counter (Number of processed operations) to 0.

On the contrary, if one of these flag is set to true, the provisioning brake counter **isn't** reset.

Setting one of the attributes **Block create operation**, **Block update operation**, **Block delete operation** from true to false will unblock the operation and reset the provisioning brake counter.

When one of the flags is set true, warning information about blocked operation is displayed on the system detail. Similar information is also displayed in the system table.



Basic information

Warning: on system are blocked some operations [Delete]!

System name

LDAP

□ Use remote connector server

After check, local connectors will not be used.

Password policy for validation

Password policy for validation

Password policy for generating

Password policy for generating

Description

□ Asynchronous provisioning

Active provisioning operations (create, update, delete), will run through the queue asynchronous running task (ProvisioningQueueTaskExecutor).

□ Read-only

Active provisioning operations create, update and delete, will not be executed.

□ Block create operation

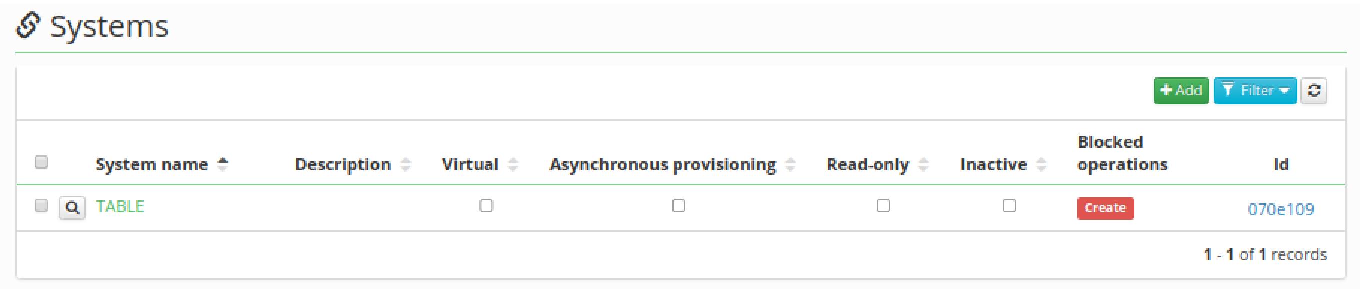
All creation operations will be blocked

□ Block edit operation

All editing operations will be blocked

☑ Block delete operation

All deletion operations will be blocked


Systems

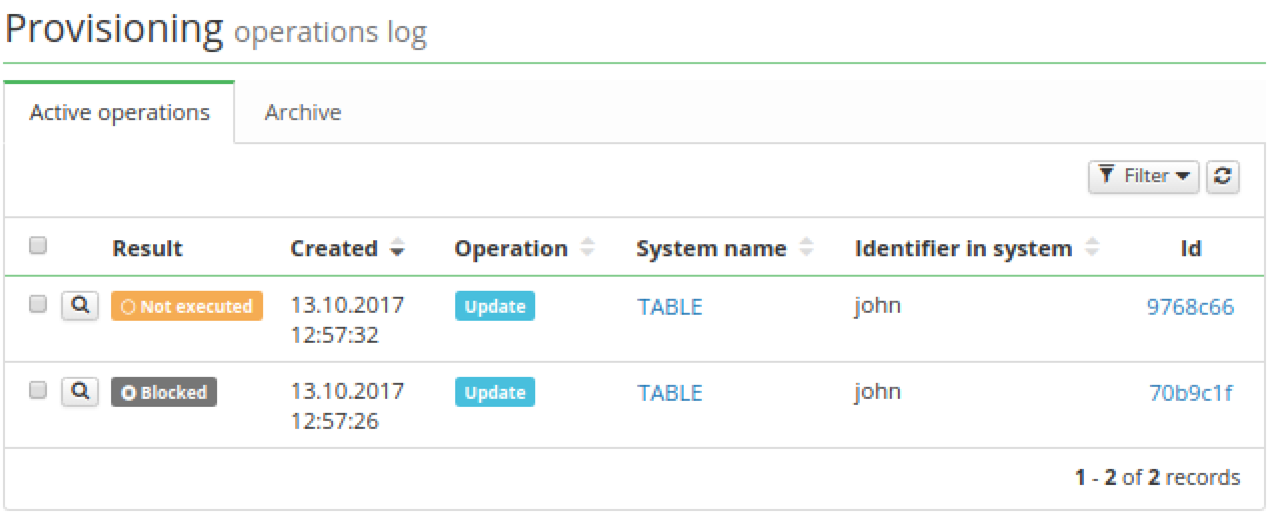
□		System name	Description	Virtual	Asynchronous provisioning	Read-only	Inactive	Blocked operations	Id
□	🔍	TABLE		□	□	□	□	Create	070e109

1 - 1 of 1 records

After restarting CzechIdM backend, the provisioning brake counter will be reset to 0. However, the flags **Block create operation**, **Block update operation**, **Block delete operation** are persistent.

Provisioning queue and blocked operation

After blocking the operation, this operation is put into the provisioning queue with the result **BLOCKED**. All the following provisioning requests of the same operation type are also put into the queue, but with the result **NOT EXECUTED**; see the queue example:



Provisioning operations log

Active operations | Archive

Filter

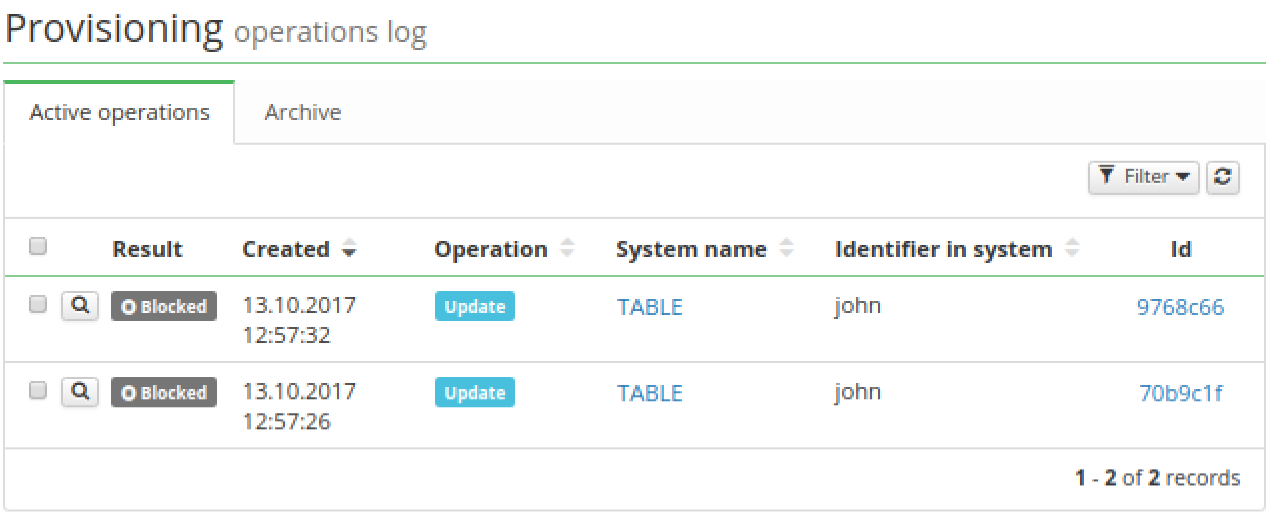
□	Result	Created	Operation	System name	Identifier in system	Id
□	○ Not executed	13.10.2017 12:57:32	Update	TABLE	john	9768c66
□	○ Blocked	13.10.2017 12:57:26	Update	TABLE	john	70b9c1f

1 - **2** of **2** records

After you unblock operation for the system, the operation counter is reset (see above), **but operations in queue aren't retried automatically.** You can decide for every provisioning request whether to retry or cancel it - see Manual retry of provisioning operations.

Unblocking the blocked operation for the system doesn't retry the operations in the queue automatically. You must run this queue manually.

If you or some process tries to retry an operation on the blocked system, this operation's result is changed from **NOT EXECUTED** to **BLOCKED**, see the picture.



Provisioning operations log

Active operations | Archive

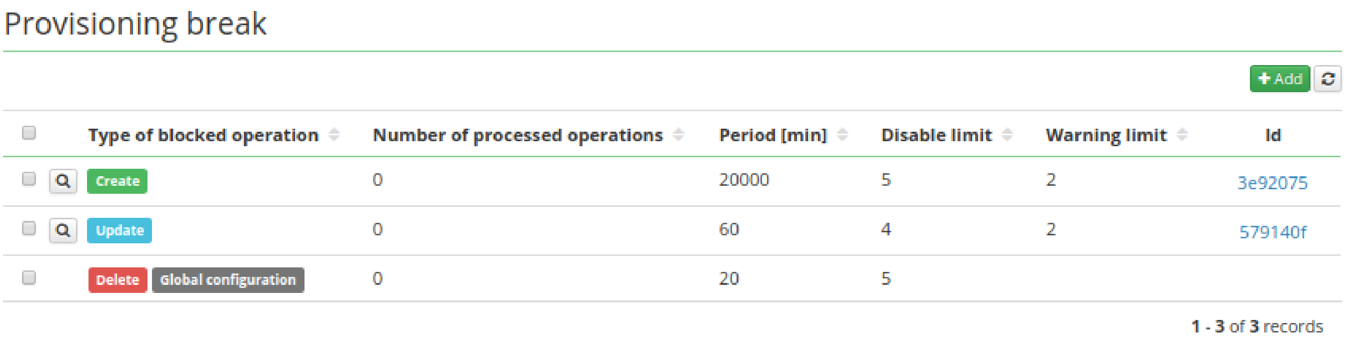
Filter

□	Result	Created	Operation	System name	Identifier in system	Id
□	○ Blocked	13.10.2017 12:57:32	Update	TABLE	john	9768c66
□	○ Blocked	13.10.2017 12:57:26	Update	TABLE	john	70b9c1f

1 - **2** of **2** records

Global provisioning brake

By application properties it's possible to set the global provisioning brake. Global provisioning brake can be defined for these types: create, update and delete. Two global provisioning brake configurations for one type are not allowed! Global provisioning brake **has lower priority** than provisioning brake configuration for specific systems. Global provisioning brake is also displayed in the provisioning brake agenda with the label **Global configuration**.



Provisioning break

□		Type of blocked operation	Number of processed operations	Period [min]	Disable limit	Warning limit	Id
□	🔍	Create	0	20000	5	2	3e92075
□	🔍	Update	0	60	4	2	579140f
□		Delete Global configuration	0	20	5		

1 - **3** of **3** records

Specific system provisioning brake configuration **overrides** global configuration.

Every global provisioning brake has these attributes:

idm.sec.provisioning.break.<TYPE>.warningLimit	If the operation exceeds this limit, the warning is added to log and also the notification is sent to all recipients.
idm.sec.provisioning.break.<TYPE>.disableLimit	If the operation exceeds this limit, the warning is added to log, notification is sent to all recipients and the system is blocked for this operation.
idm.sec.acc.provisioning.break.<TYPE>.period	For this period the limits are evaluated. The value is in minutes.
idm.sec.acc.provisioning.break.<TYPE>.templateWarning	Template that will be sent after exceeding the warning limit.
idm.sec.acc.provisioning.break.<TYPE>.templateDisable	Template that will be sent after exceeding the disable limit.
idm.sec.acc.provisioning.break.<TYPE>.identityRecipients	Identity usernames or IDs split by comma
idm.sec.acc.provisioning.break.<TYPE>.roleRecipients	Role codes or IDs split by comma
idm.sec.acc.provisioning.break.<TYPE>.disabled	Boolean flag if configuration is disabled (Inactive).

Note that the global configuration defines recipients as ID or code (there is used lookup service).

Example configuration of a global delete provisioning brake:

```
idm.sec.acc.provisioning.break.delete.warningLimit=2
idm.sec.acc.provisioning.break.delete.disableLimit=5
idm.sec.acc.provisioning.break.delete.period=20
idm.sec.acc.provisioning.break.delete.templateWarning=warningTemplateForProv
isioningBreak
idm.sec.acc.provisioning.break.delete.templateDisable=disableTemplateForProv
isioningBreak
```

```
idm.sec.acc.provisioning.break.delete.identityRecipients=admin,owner
idm.sec.acc.provisioning.break.delete.roleRecipients=exampleRole,systemAdministrators
idm.sec.acc.provisioning.break.delete.disabled=false
```

From:
https://wiki.czechidm.com/ - **CzechIdM Identity Manager**

Permanent link:
https://wiki.czechidm.com/tutorial/adm/provisioning

Last update: **2018/12/28 12:26**