Modules - vs: VS account lifecycle and module configuration

Basic life cycle of virtual system accounts

We have a **john.doe** identity and a role **RoleVS** linked to a virtual system **VS**.

1. We assign the role 'RoleVS' to the identity **john.doe**. It will trigger provisioning for the connected systems including the system 'VS'.
2. The virtual system 'VS' creates a request for creating a new account 'john.doe'. It sends notifications to all **implementers** - users configured in the settings of this virtual system.
3. The identity 'john.doe' has now an account on the virtual system, but this account is not yet confirmed by the implementers.
4. We'll make a change on our account when we edit the last name of the identity from **Doe** to **Roe**.
5. Saving the identity will trigger provisioning for the connected systems.
6. Because the account 'john.doe' on the virtual system already exists (although only in unconfirmed request form), provisioning evaluates that it's needed to edit the **lastName** attribute to the new value **Roe**.
7. The virtual system creates a second record in the VS request agenda and sends notifications to all implementers.
8. Implementers have now two notifications and two corresponding requests in IdM. The first is for creating and the second for changing the same account. In the second notification, there is also information about the previous unresolved request.
9. **The implementer makes a manual change on the target system - creates the account and changes its last name.**
10. The implementer confirms in the Requests agenda in IdM that the account was created.
11. By confirming the request, the creation of the account 'john.doe' is written into the VS data structure (a new item in VsAccount).
12. The implementer confirms Requests agenda in IdM that the last name of the account was changed.
13. By confirming the request, the change (**Doe** → **Roe**) is written into the VS data structure.

How to create a virtual system

- In the left menu, select 'Virtual systems / List' and click on Add.

It displays a dialog to create a new virtual system.

You can fill:

- Name for the virtual system, e.g. 'NewVirtualSystem'
- Implementers - users, who will receive the requests for updating the real system

- Roles of implementers - users who have assigned these roles will be receiving requests for updating the real system

Beware: Users/roles have to have permission '**Requests on virtual systems (VsRequest)**' to receive these requests.

When the configuration of implementers of a virtual system is changed (either the set of implementers or implementer's roles are changed), this change has to be propagated to the configuration of the VS module. The propagation can be triggered manually by clicking on the **Test** button on the virtual system configuration tab. Also, any following operation on the virtual system will also trigger the propagation, so new requests will be created with the actual configuration of the implementers.

In the detail of the new virtual system, the system schema, mapping and attributes are configured by default

Create a new role

We have created the new virtual system. Now we will assign the system to some users. For this we create a new role and create the mapping for our new virtual system.

- In the left menu, select Roles. Click on 'Add' green button to create the new role.
- You have to only fill the name for your new role, e.g. 'RoleForNewVirtualSystem'.
- Click on 'Save and continue'.

Create the mapping on the virtual system

- In the detail of our newly created role, select the tab 'Systems'
- Click on 'Add' green button.
- In 'System' field, select our virtual system, on picture it is 'NewVirtualSystem'.
- In 'Mapping' field, select 'Default provisioning (Identity - Provisioning)'.
- and click on 'Save'

Create a new user

We will create a new user and assign him our role, so he will be provisioned to our new virtual system.

- In left menu, select 'Users'
- Click on 'Create user' green button
- Fill Login, First name and Surname (e.g. 'john.doe', 'John', 'Doe').
- Click on 'create and edit'.
- On the user detail, click on the tab 'Roles'.
- Click on 'Manage Authorizations'.
- On the displayed dialog will be added the new role. Click on 'Add' green button.
- In the field 'Role name' select our role 'RoleForNewVirtualSystem'.
- Click on 'Set'.
- Click on 'Submit a request'.

Requests

The Requests agenda of the Virtual Systems ('**Virtual systems / Requests**') contains all operations, that were requested by IdM on virtual systems. Every **request** contains the identification of the account and the virtual system, the operation that should be done with the account (create/update/delete), the list of implementers who should implement the request, etc.

The Requests agenda is divided into two tabs:

- Unresolved requests of the logged user. This list contains all virtual system requests that haven't been implemented and confirmed yet.
- The Archive. This list contains all resolved, duplicated or canceled requests.

In our example, the implementers received a new task to create a new account 'john.doe' on the virtual system 'NewVirtualSystem'. To view the request, go to '**Unresolved requests**' in '**Virtual systems / Requests**'.

You can go to the detail of the request with UID 'john.doe' and the system 'NewVirtualSystem' (click on the button with "magnifying glass"). You can now see the detail of the request for creating new account.

There are three specific groups of information:

- **Basic information** for the request (state, UID, system, type, created).
- **Implementers** - All implementers who can resolve the request. The list includes all directly selected implementers and all users with assigned implementer's roles (Implementers can be configured during the creation of the virtual system or in the detail of the virtual system).
- **Target state on system** - The table displays how the account should be set on the target system. There are only changes from the request (not from previous/next unresolved requests). For the 'create' request types, there are all rows marked as changed (orange color). Generally, this table shows the state in CzechIdM against the state on the virtual system.

If we do not resolve the create request and edit our user '**john.doe**', e.g. change user's surname to Doe. A new update request is in '**Unresolved requests**'. Click on the detail of the update request.

- Request detail displays the same information as in the previous 'create' case, just for the single attribute. Because previous 'create' request was not confirmed yet, is shows only one row. If the 'create' request is resolved, then it shows all attributes of the account with one changed attribute.
- Request detail can display previous and next unresolved request (for the same account). For this case it shows the table with one previous request (our known 'create' request).

When we finally resolve our two requests, they are moved to the tab '**Archive**'.

Operations with the request

Implement request

The Implement operation marks the request as '**Implemented**'.

- The operation '**Implemented**' can be started in the table '**Unresolved requests**' or in the request detail.
- After this operation is executed, all changes from this request are propagated to the data structure of the virtual system.
- The request is moved to '**Archive**'

When a 'delete' request is implemented and there are some unresolved previous requests for the same account, all the previous requests are automatically canceled.

Cancel request

The Cancel operation marks the request as '**Canceled**'.

- Operation '**Cancel**' can be started in the table '**Unresolved requests**' or in the request detail.
- It is required to fill the reason for this operation. The reason is displayed in the archived request detail.
- After this operation is executed, all changes from this request are discarded.
- All unresolved requests made after this canceled request on the same account are canceled as well. Fix Me! check this, is it true?
- The request is moved to '**Archive**'

Permissions

- '**VsRequest**' permission is required for displaying the request agenda.
- For displaying requests only for assigned implementers, you have to set evaluator '**VsRequestByImplementerEvaluator**'. When setting this evaluator, users can view and edit requests where they are implementers (directly or by assigned roles).

Notifications

After the request for updating virtual system is created, the notification is sent to all implementers.

As default was implemented email notification '**vs:vsRequestCreated**' for new virtual system requests. This notification is by default automatically connected to virtual systems requests. The email template for this notification can be modified in left main menu '**Notifications / Templates**'.

Email template provides similar information as the request detail (described above). For example, 'Target table' is constructed from same data as the table on the request detail. See below for examples of notification emails that are sent during the process described in its basic live cycle (creating and modifying the account 'john.doe') and notification, which displays the change of a multivalued attribute.

Email notification for creating a new account 'john.doe':

Email notification for modifying the account 'john.doe':

Email notification for modifying the multivalued attribute 'ldapGroups' for the account 'john.doe'.

New values 'E,F' were added to the attribute and the values 'C,B' were removed:

Virtual systems configuration

BasicVirtualConfiguration contains these attributes:

- **Required confirmation** (boolean) - If not checked, then all requests for this virtual system will be resolved immediately (this will be visible on archived requests). No notification will be sent to implementers. The confirmation is required by default.
- **Attributes** (List of String) - Properties for create EAV model. This list defines the "columns" for the virtual system in the internal IdM repository. VsAccount contains only UID and Enable static columns. Others columns must be defined in the EAV model. By default, the basic attributes

from Identity are set (firstName, lastName, email, titleAfter, titleBefore, phone).
- **Implementers** (List of UUID) - Users, who will receive the requests for updating the real system. Every implementer must be an identity in CzechIdM. Values are identifiers of identities.
- **Roles of implementers** (List of UUID) - Users who have assigned these roles will be receiving requests for updating the real system. Every role must be a role in CzechIdM. Values are identifiers of roles.
- **Supports account disable/enable** (boolean) - If checked, then accounts disabling/enabling is supported. It means, that in the schema will be generated the attribute '_ _ ENABLE _ _'.

Default implementers

If the implementers of the virtual system are not directly configured, the default implemeter's role will be used when creating the virtual systems requests. The default role can be defined in the IdM configuration:

```
idm.sec.vs.role.default=<some-code-of-role>
```

The default value of the default implementer's role is **superAdminRole**.