Attributes of role

role, attributes, parameters, assigned, parametrization

What are role attributes

Role attributes determine what additional information **can (must)** be filled in a user's **assigned role**. A typical example can be the **IP address** of a user's end station, which must be filled in a role assignment request.

The **definitions of what attributes** are to be filled for the role are managed on the **role detail** (role attributes tab). Here, you define not only what attributes to show in the request, but also their **default values** and **validation settings**. This definition is part of the **role approval** process (off by default).

How it works

Definition

First, you need to **create the main definition** of all attributes which could be used in role **sub-definition**. The **main form definition** can be created in the **Form definitions** agenda.



Attributes from the **main definition** can be used multiple times in multiple roles.



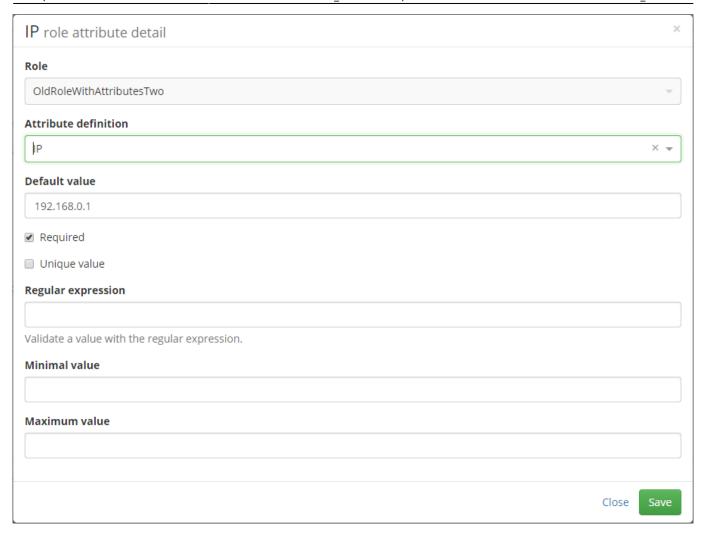
Main definition must be created for type **IdmIdentityRole**.



Now isn't supported confidential and attachment attributes!

Then, you need to **create sub-definition** of attributes witch should be filled in requesting a role. Sub-definition can be created on the role detail on the **Role attributes** tab. In sub-definition you can select an attributes only from main definition. So first what you need to do, is select main definition and save it. Then you can create **attribute definition for the role**.





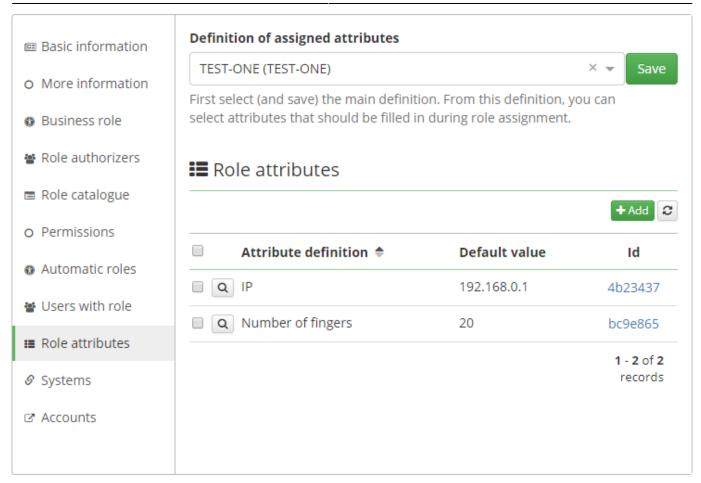
Attribute definition for the role define:

* **Definition of attribute** from the main definition. * **Overrides the default value**. Default value from attribute in main definition will be prefilled (only on frontend). * **Overrides the validation settings**. Validation settings from attribute in main definition will be prefilled (only on frontend).



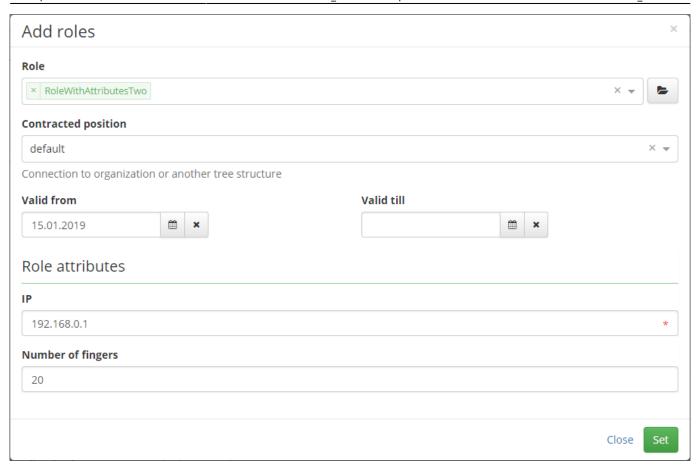
Only attributes defined in the sub-definition will be show on role requesting!

2025/04/19 05:58 3/9 Attributes of role



Using on the role request

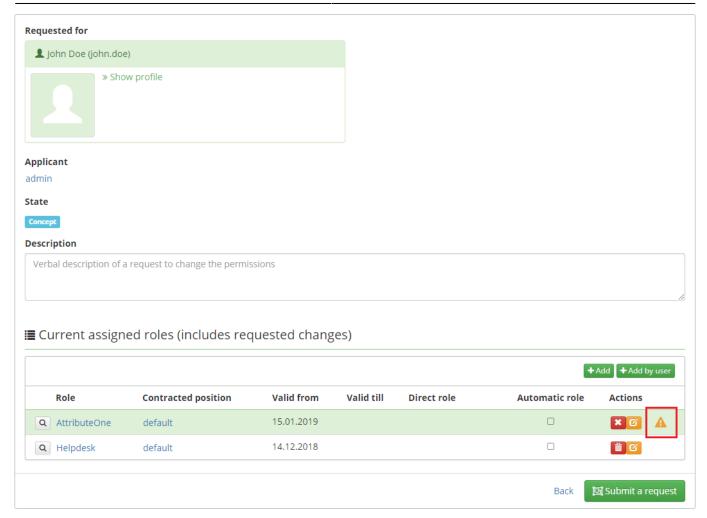
If some requested role has attributes, then is rendered form on role concept detail:



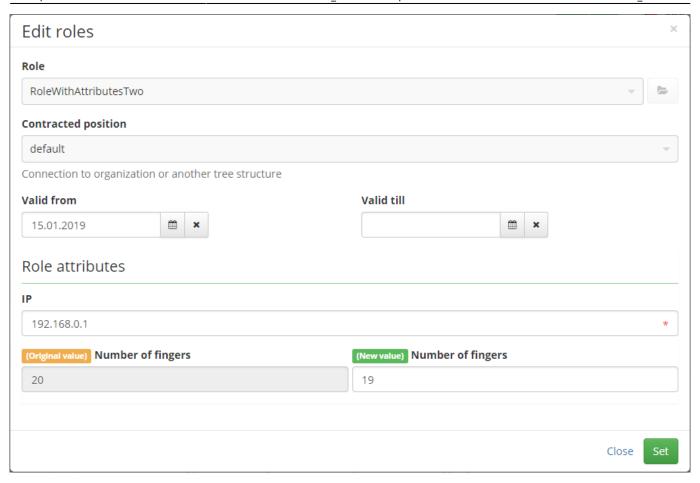


Form with role's attributes is rendered only if **one role is selected**! If are selected more roles on the detail, then user **cannot fill the attribute's values** directly. In this case user can create role-concepts for more roles and then editing the values of attributes for each concept **one by one**.

If you add multiple roles in one request and some from this roles will have required attribute (without default value), then will be created concept **not valid**. In this case you will see **warning icon** on the unvalid concept.

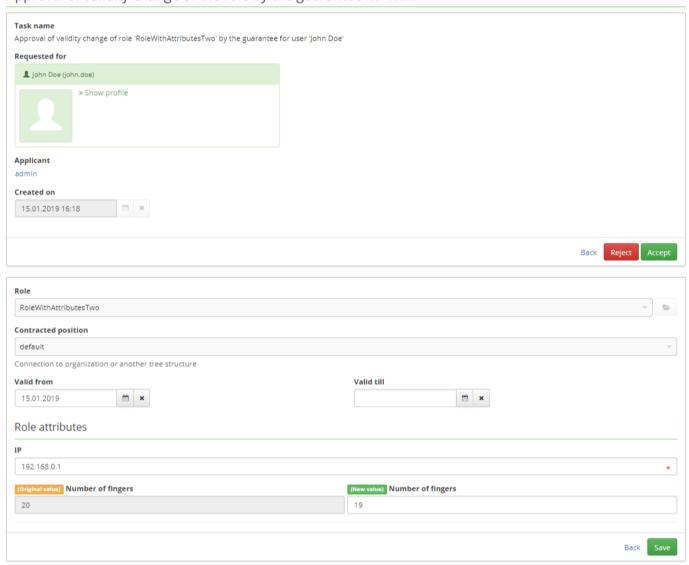


You can **modified existing attribute value** in assigned role. For this case was created new mode for **highlight** changes on detail of role concept:

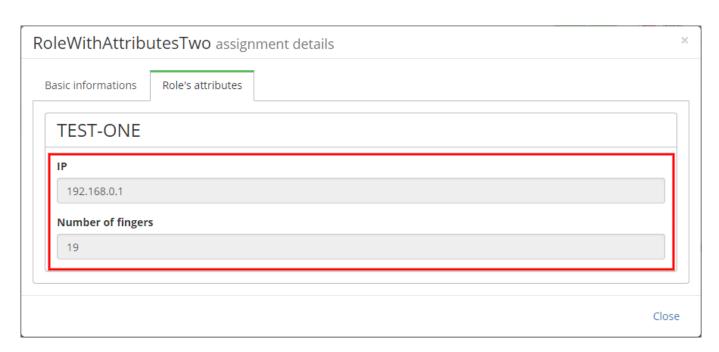


If a **request** that contains **attribute roles** is submitted, a standard **approving process is executes**. If the role with the attributes is approval, then the detail of the concept role, including the attributes, **is displayed in the detail of the approval user task**.

Approval of validity change of the role by the guarantee Task details



Upon successful **completion of the request**, the resulting concept role attributes are **copied** to the **assigned role** (**IdmIdentityRole**). Attributes assigned to roles can be displayed on the **identity detail** (**Roles** \rightarrow **detail** \rightarrow **Tab** "**Role's attributes**").



Provisioning of assigned roles

For the purpose of provisioning assigned roles, new attributes (**User assigned roles** and **User assigned roles** (**for this system**)) were created, which can be used in system mapping for provisioning identities. Input of the transformation into the system is a list of valid assigned identity roles. This assignment is represented by the **AssignedRoleDto** object, which mirrors the object **IdmIdentityRoleDto** and tries to simplify the work with assigned roles in the transform. This simplification is primarily based on the fact that the object contains the entire DTO (role, identityContract, ...) instead of the UUID and mainly contains a **map of all the attributes of the assigned role** (where the key is the attribute code and the value is a list of all attribute values).



Input for transforming into the system (attributeValue) is a list of all valid user roles (list of AssignedRoleDto). This attribute is only for provisioning and for default strategy (SET).



Structure of **AssignedRoleDto** is here.



Input of transformation for attribute **User assigned roles (for this system)** will contain only valid assigned user roles assigned this system!

Example script for print assgined roles to the string:

```
import eu.bcvsolutions.idm.acc.domain.AssignedRoleDto;
import eu.bcvsolutions.idm.core.api.dto.IdmRoleDto;

if (attributeValue) {
    String result = "";
    for(AssignedRoleDto dto : attributeValue) {
        result = result + "Role: [" + dto.getRole().getCode() + "]

Attributes: [" + dto.getAttributes().toString() + "]";

    }
    return result;
}
return "";
```

Output of this script:

```
Role: [Helpdesk] Attributes: [[:]]
Role: [RoleWithOneAttribute] Attributes: [[IP:[192.168.0.122]]]
Role: [RoleWithTwoAttributes] Attributes: [[Number of fingers:[20],
IP:[168.192.0.10]]]
```

2025/04/19 05:58 9/9 Attributes of role

Admin guide (to be completed)

• Role attributes

From:

https://wiki.czechidm.com/ - IdStory Identity Manager

Permanent link:

https://wiki.czechidm.com/devel/documentation/role_attributes

Last update: 2019/04/16 08:07

