

# Relation Configuration



## MidPoint 3.9 and later

This feature is available only in midPoint 3.9 and later.

**Relation** is an important mechanism that is used at many places in midPoint. But perhaps the most important usage is to enable advanced features of **RBAC** and **organizational structure management**. Older midPoint versions had hardcoded set of relations that could not be customized. MidPoint version 3.9 introduced partial configuration of relations. Now it is possible to add new relation that will be used by midPoint in addition to hardcoded relations.

The relations are configured in **system configuration object**:

```
<systemConfiguration oid="00000000-0000-0000-0000-000000000001"
  xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
  xmlns:org="http://midpoint.evolveum.com/xml/ns/public/common/org-3"
  xmlns:piracy="http://midpoint.evolveum.com/xml/ns/samples/piracy">
  ...
  <roleManagement>
    <relations>
      <relation>
        <ref>piracy:captain</ref>
        <description>This is completely new relation</description>
        <display>
          <label>Captain</label>
        </display>
        <category>organization</category>
        <category>governance</category>
      </relation>
      <relation>
        <ref>org:owner</ref>
        <description>This is redefined default relation. EXPERIMENTAL</description>
        <display>
          <label>Master</label>
        </display>
        <category>policy</category>
        <category>governance</category>
        <defaultFor>owner</defaultFor>
      </relation>
    </relations>
  </roleManagement>
</systemConfiguration>
```

The configuration above is adding one new relation to the system: `captain`. This relation will work in the same way as hardcoded relations, but it will not have any special functionality that is associated with special relations such as `deputy`.

It is recommended to use your own custom namespace for custom relations. Such as the `piracy` namespace in the example above. End user will not see the namespace at all, it is just an internal mechanism. It is likely that new built-in relations will be introduced in future midPoint versions. Using separate namespaces is a mechanism to avoid identifier conflict in future midPoint versions.

Relation can be sorted into **categories** categories. Each category determines in which parts of the user interface will be particular relation used. See **User Interface Area Categories** page for more details.

Currently, relation configuration is supposed to be used only to add completely new relations. Changing existing (hardcoded) relations is **experimental functionality**.

## Relation Behavior

Since 3.9, midPoint allows to completely redefine object relations. Instead of specific relation *names* midPoint defines behavior depending on relation *kinds*. The following kinds are available (see **RelationKindType**).

Relation kind	Meaning	Default relations that are of this kind
---------------	---------	---

member	<p>Membership relation, usually meaning "has" or "is member of". Specifies that the subject is a member of organization, or that the subject has been assigned a role in a way that he gets authorizations and other content provided by that role.</p> <p>Default relation of <code>member</code> kind is also considered to be the overall default relation (i.e. used when reference relation is null).</p>	org:default, org:manager
manager	<p>Relations of "is manager of" kind. Specifies that the subject is a manager of organizational unit. Relations of this kind are usually also of <code>member</code> kind.</p>	org:manager
meta	<p>Relations used for metarole assignments. Sometimes it is important to distinguish metarole and member assignments. This kind of relation is used for that purpose.</p>	org:meta
delegation	<p>Relation of "is deputy of" kind. Specifies that the subject is a deputy of another user.</p>	org:deputy
approver	<p>Relation "is approver of" kind.</p> <p>Specifies that the subject is a (general) approver of specified (abstract) role. The approver will be asked for decision if the role is assigned, if there is a rule conflict during assignment (e.g. SoD conflict) or if there is any similar situation.</p> <p>This approver is responsible for the use of the role, which mostly means that he decides about role assignment. It is NOT meant to approve role changes. Role owner is meant for that purpose.</p>	org:approver
owner	<p>Relation of "is owner of" kind.</p> <p>Specifies that the subject is a (business) owner of specified (abstract) role. The owner will be asked for decision if the role is modified, when the associated policy changes and so on.</p> <p>This owner is responsible for maintaining role definition and policies. It is NOT necessarily concerned with role use (e.g. assignment). The <code>approver</code> relation kind is meant for that purpose.</p>	org:owner
consent	<p>Relation "is consent for" kind. Specifies that the subject gave a consent for using personnel information related to this role.</p>	org:consent

Note that a relation can be of more than one kind. For example, `org:default` is of `member` and `manager` kinds.

## Hardcoded Relations

There is a handful of relations that are hardcoded in midPoint:

Relation	Meaning	Is a default for	Is also of kind
org:default	<p>Default relation, usually meaning "has" or "is member of". Specifies that the subject is a member of organization, or that the subject has been assigned a role in a way that he gets authorizations and other content provided by that role.</p>	member	-
org:manager	<p>Relation "is manager of". Specifies that the subject is a manager of organizational unit.</p>	manager	member
org:meta	<p>Relation used for metarole assignments. Sometimes it is important to distinguish metarole and member assignments. This relation is used for that purpose.</p>	meta	-
org:deputy	<p>Relation "is deputy of". Specifies that the subject is a deputy of another user.</p>	delegation	-
org:approver	<p>Relation "is approver of". Specifies that the subject is a (general) approver of specified (abstract) role. The approver will be asked for decision if the role is assigned, if there is a rule conflict during assignment (e.g. SoD conflict) or if there is any similar situation.</p> <p>This is a generic approver used for all the situation. The system may be customized with more specific approver roles, e.g. <code>technicalApprover</code>, <code>securityApprover</code>, etc.</p> <p>This approver is responsible for the use of the role, which mostly means that he decides about role assignment. It is NOT meant to approve role changes. Role owner is meant for that purpose.</p>	approver	-
org:owner	<p>Relation "is owner of". Specifies that the subject is a (business) owner of specified (abstract) role. The owner will be asked for decision if the role is modified, when the associated policy changes and so on.</p> <p>This owner is responsible for maintaining role definition and policies. It is NOT necessarily concerned with role use (e.g. assignment). The <code>approver</code> relation is meant for that purpose.</p>	owner	-
org:consent	<p>Relation "is consent for". Specifies that the subject gave a consent for using personnel information related to this role.</p>	consent	-

Meaning of these statically defined relation are defined directly within midPoint code. Before midPoint 3.9 this set of relations was effectively fixed. Since midPoint 3.9 this can be extended and even changed. Just please note that currently relation configuration is supposed to be used only to add completely new relations. Changing existing (hardcoded) relations is [experimental functionality](#).

## See Also

- [Relation](#)
- [Advanced Hybrid RBAC](#)
- [Organizational Structure](#)
- [System Configuration Object](#)
- [Relation Repository](#)