

Non-tolerant induced focus mappings HOWTO (experimental)

⚠ EXPERIMENTAL

This feature is **experimental**. Although it should work, please use at your own risk. This feature is not covered by midPoint support. In case that you are interested in [supporting](#) development of this feature, please consider purchasing [midPoint Platform subscription](#).

ℹ MidPoint 3.7.1 and later

This feature is available only in midPoint 3.7.1 and later.

There are situations when you need to create non-tolerant focus mappings. This means that the focus (e.g. user) should have no values in particular item, unless provided by these mappings.

For mappings residing in object template this feature has been available since midPoint 3.3 (although marked as experimental); but until now it was not available for mappings induced by assigned roles. Here we show how to do that.

Example

Let us consider the example mentioned [here](#) (copied here for simplicity):

We "stamp" each focus with the description of assigned role.

To be effective, we outsource logic to one common meta role that all other stamping roles assigns:

Stamping metarole

```
<role oid="8ce0a59d-dd67-40b4-b2d3-80f9173afa9c">
  <name>Metarole: Stamping service</name>
  <inducement>
    <focusMappings>
      <mapping>
        <expression>
          <script>
            <code>
              basic.stringify(assignmentPath[0].target.description) + '-bearer'
            </code>
          </script>
        </expression>
        <target>
          <path>fullName</path>
        </target>
      </mapping>
    </focusMappings>
    <order>2</order>
    <focusType>UserType</focusType>
  </inducement>
</role>
```

Then each stamping role has assignment to this metarole:

Stamping role

```
<role>
  <name>Stamping role</name>
  <description>stamp1</description>
  <assignment>
    <targetRef oid="8ce0a59d-dd67-40b4-b2d3-80f9173afa9c" type="RoleType">
      <!-- Metarole: Stamping service -->
    </targetRef>
  </assignment>
</role>
```

Finally, any UserType that has assigned the Stamping role will have its fullName attribute populated with the value of the Stamping role's description ("stamp1-bearer" in this case).

In order to remove fullName attribute value when no stamping role is present, we create the following code in the user template:

Code to fetch values computed by assignment mappings

```
<item>
  <ref>fullName</ref>
  <tolerant>false</tolerant>
  <mapping>
    <expression>
      <script>
        <code>
          import com.evolveum.midpoint.prism.path.*
          def vals = midpoint.collectAssignedFocusMappingsResults(new ItemPath("fullName"))
          vals.isEmpty() ? null : vals.iterator().next().realValue
        </code>
      </script>
    </expression>
    <evaluationPhase>afterAssignments</evaluationPhase>
  </mapping>
</item>
```

The mapping marks fullName attribute as non-tolerant. It means that all values not provided by the object template mappings are discarded. And we provide a very special mapping in the object template: it calls (experimental) method `midpoint.collectAssignedFocusMappingsResults` which inspects recently computed assignments (`lensContext.evaluatedAssignmentTriple`), takes all non-negative values from mappings relevant to fullName from all evaluated non-negative assignments, and returns the value (if present) as its output.

Note: if the target attribute is multivalued, the following template mapping should be used (it should work for the single-valued case as well):

Code to fetch values computed by assignment mappings - to multivalued item (employeeType in this case)

```
<item>
  <ref>employeeType</ref>
  <tolerant>false</tolerant>
  <mapping>
    <expression>
      <script>
        <relativityMode>absolute</relativityMode>
        <code>
          import com.evolveum.midpoint.prism.path.*
          midpoint.collectAssignedFocusMappingsResults(new ItemPath("employeeType"))
        </code>
      </script>
    </expression>
    <evaluationPhase>afterAssignments</evaluationPhase>
  </mapping>
</item>
```