

Task chaining HOWTO



The approaches described here are somewhat experimental. They might or might not work in any particular case.

Imagine you want to ensure that Task1 will execute before Task2; and that Task2 will start immediately after Task1 finishes.

There are two options here. None of them is officially supported yet. But they usually work, with some limitations.

Option 1: Using simple task dependencies

It works as this:

- the dependent Task2 will not start unless the first Task1 is either closed or deleted
- the result (fatal or partial failure, success) is not taken into account

So, basically:

1. closing or deleting the first Task1 -> the second Task2 will start [when deleting it might take some time until midPoint realizes the first one has disappeared]
2. suspending the first Task1 -> the second Task2 will not start
 - a. of course, after you close, delete the first Task1, the second Task2 will start

This could work:

Step 1: prepare the following two task objects:

First task

```
<task xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
      xmlns:q="http://prism.evolveum.com/xml/ns/public/query-3"
      xmlns:c="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
      oid="63fa34c4-0498-42a9-99ee-88506c79e975"
      version="1">
  <name>Recompute users</name>
  <extension xmlns:mext="http://midpoint.evolveum.com/xml/ns/public/model/extension-3"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="c:ExtensionType">
    <mext:objectType>c:UserType</mext:objectType>
  </extension>
  <taskIdentifier>1494860532840-0-1</taskIdentifier>
  <ownerRef xmlns:org="http://midpoint.evolveum.com/xml/ns/public/common/org-3"
            oid="00000000-0000-0000-0000-000000000002"
            relation="org:default"
            type="c:UserType"><!-- administrator --></ownerRef>
  <dependent>1494860531232132-0-1</dependent>
  <executionStatus>suspended</executionStatus>
  <category>Recomputation</category>
  <handlerUri>http://midpoint.evolveum.com/xml/ns/public/model/synchronization/task/recompute/handler-3<
/handlerUri>
  <recurrence>single</recurrence>
</task>
```

Second task

```
<task xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
  xmlns:c="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
  oid="465fae8e-8229-4ad8-8c95-9d49fa589b35"
  version="1">
  <name>Recompute roles (after users)</name>
  <extension xmlns:mext="http://midpoint.evolveum.com/xml/ns/public/model/extension-3"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:type="c:ExtensionType">
    <mext:objectType>c:RoleType</mext:objectType>
  </extension>
  <taskIdentifier>1494860531232132-0-1</taskIdentifier>
  <ownerRef xmlns:org="http://midpoint.evolveum.com/xml/ns/public/common/org-3"
    oid="00000000-0000-0000-0000-000000000002"
    relation="org:default"
    type="c:UserType"><!-- administrator --></ownerRef>
  <executionStatus>waiting</executionStatus>
  <waitingReason>otherTasks</waitingReason>
  <category>Recomputation</category>
  <handlerUri>http://midpoint.evolveum.com/xml/ns/public/model/synchronization/task/recompute/handler-3<
/handlerUri>
  <recurrence>single</recurrence>
</task>
```

Note that the first task is suspended and the second one is in waiting state.

Step 2: Import the tasks.

Step 3: Resume the first task.

Now the first task should execute. After it's closed the second one should start and execute.

The disadvantage of this solution is that re-execution of the tasks requires the second task to be manually switched from `closed` state back to `waiting / otherTasks` one.

Option 2: Using partitioned tasks

This option is available since midPoint 3.8. This version brought so called [Partitioned tasks](#). Although this feature is primarily used for partitioning standard tasks (like Reconciliation or Validity scanning), nothing prevents us from using it for custom tasks as well.

The following composite task prepares a CSV file and imports it.

```

<task oid="8f8de5ad-e699-439e-8362-77cbb994117c"
  xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
  xmlns:c="http://midpoint.evolveum.com/xml/ns/public/common/common-3"
  xmlns:t="http://prism.evolveum.com/xml/ns/public/types-3"
  xmlns:se="http://midpoint.evolveum.com/xml/ns/public/model/scripting/extension-3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:noop="http://midpoint.evolveum.com/xml/ns/public/task/noop/handler-3"
  xmlns:ri="http://midpoint.evolveum.com/xml/ns/public/resource/instance-3"
  xmlns:org="http://midpoint.evolveum.com/xml/ns/public/common/org-3">
  <name>Prepare and import CSV file</name>
  <extension xmlns:mext="http://midpoint.evolveum.com/xml/ns/public/model/extension-3" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance" xsi:type="c:ExtensionType">
  <!-- This extension is copied to both partitions (subtasks). Each one takes items that are relevant to
it. -->
  <mext:kind>account</mext:kind>
  <mext:objectclass>ri:AccountObjectClass</mext:objectclass>
  <se:executeScript xmlns:s="http://midpoint.evolveum.com/xml/ns/public/model/scripting-3">
    <s:action>
      <s:type>execute-script</s:type>
      <s:parameter>
        <s:name>script</s:name>
        <c:value xsi:type="c:ScriptExpressionEvaluatorType">
          <c:code>
            File file = new File("C:/tmp/file.csv")
            file.write "ident,number,firstname,lastname\nferko,11,Ferko,Mrkvicka\njanko,12,
Janko,Novak"
          </c:code>
        </c:value>
      </s:parameter>
      <s:parameter>
        <s:name>forWholeInput</s:name>
        <c:value>>true</c:value>
      </s:parameter>
    </s:action>
  </se:executeScript>
</extension>
<ownerRef oid="00000000-0000-0000-0000-000000000002"/>
<objectRef oid="ef2bc95b-76e0-48e2-86d6-3d4f02d3fafe" relation="org:default" type="c:ResourceType"/>
<executionStatus>runnable</executionStatus>
<handlerUri>http://midpoint.evolveum.com/xml/ns/public/task/generic-partitioning/handler-3</handlerUri>
<workManagement>
  <taskKind>partitionedMaster</taskKind>
  <partitions>
    <copyMasterExtension>true</copyMasterExtension>
    <partition>
      <index>1</index>
      <taskName>Prepare CSV</taskName>
      <handlerUri>http://midpoint.evolveum.com/xml/ns/public/model/scripting/handler-3</handlerUri>
    </partition>
    <partition>
      <index>2</index>
      <taskName>Import CSV</taskName>
      <handlerUri>http://midpoint.evolveum.com/xml/ns/public/model/synchronization/task/import
/handler-3</handlerUri>
    </partition>
  </partitions>
</workManagement>
<recurrence>single</recurrence>
</task>

```

There are some limitations, though. For example, if any of the subtasks end in a failure, the processing continues - and the overall result is "OK" even if subtasks fail. So, to see the status in an accurate way, you have to display subtasks along with root tasks.

The advantage is that the re-execution of such composite task is quite simple. It can be done either manually, or you can even make the master task recurring.