

Managing midPoint Resource Configuration Files

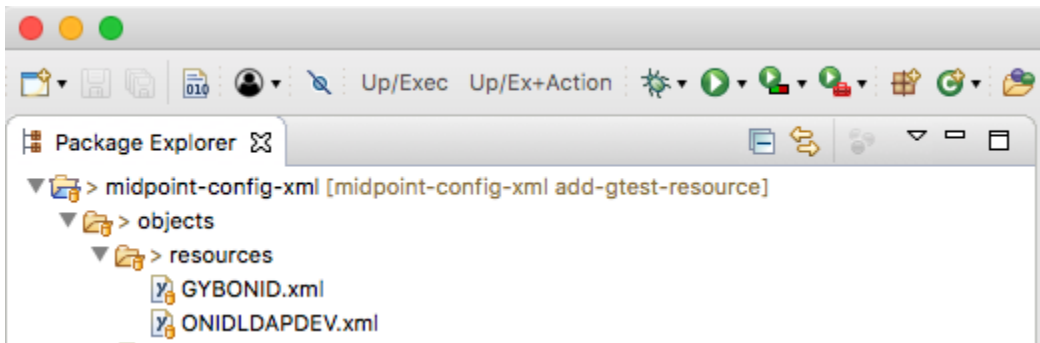
midPoint configuration is maintained in XML files by default. This post describes how Oregon State University is managing our midPoint resource configuration files by leveraging git for storage and change tracking, and Eclipse as an IDE for editing the configuration and applying it to the midPoint server.

Assumptions

- You are running midPoint 3.6+
- You are familiar with the basic operation of git
- You have git and Eclipse installed

midPoint Eclipse Plugin

1. Install the midPoint Eclipse Plugin by following Evolveum's [Eclipse Plugin Installation HOWTO](#).
2. Configure the plugin to connect to your midPoint server. See the "Configuring Connections" section in Evolveum's [Eclipse Plugin HOWTO](#).
3. Create a new project in Eclipse with a type of "General". This project will contain the midPoint configuration files. In this guide, our project is called "midpoint-config-xml".



Git Repository Layout

The git repository layout for our midPoint configuration files follows the folder structure created by the midPoint Eclipse plugin. Here is an example repo layout:

```
.
├── README.md
├── objects
│   └── resources
│       ├── GYBONID.xml
│       └── ONIDLDAPDEV.xml
```

GYBONID and ONIDLDAPDEV are midPoint resources, and their configurations are maintained in the corresponding XML files.

Tutorial: Adding a New Resource

In this tutorial, you will add a new resource in midPoint for Google Apps, use Eclipse to make changes to its configuration, and use git to store those changes.

Suppose you have a midPoint configuration project in Eclipse that points to files stored in the following git repo:

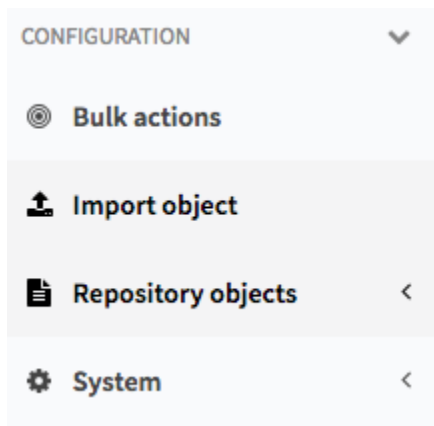
```
midpoint-config-xml
├── objects
│   └── resources
│       ├── GYBONID.xml
│       └── ONIDLDAPDEV.xml
```

Step 1. Download sample resource XML

Most midPoint connectors provide a sample resource XML file to use as a configuration starting point. Click here to [download the Google Apps sample XML](#).

Step 2. Import the sample resource XML file into midPoint

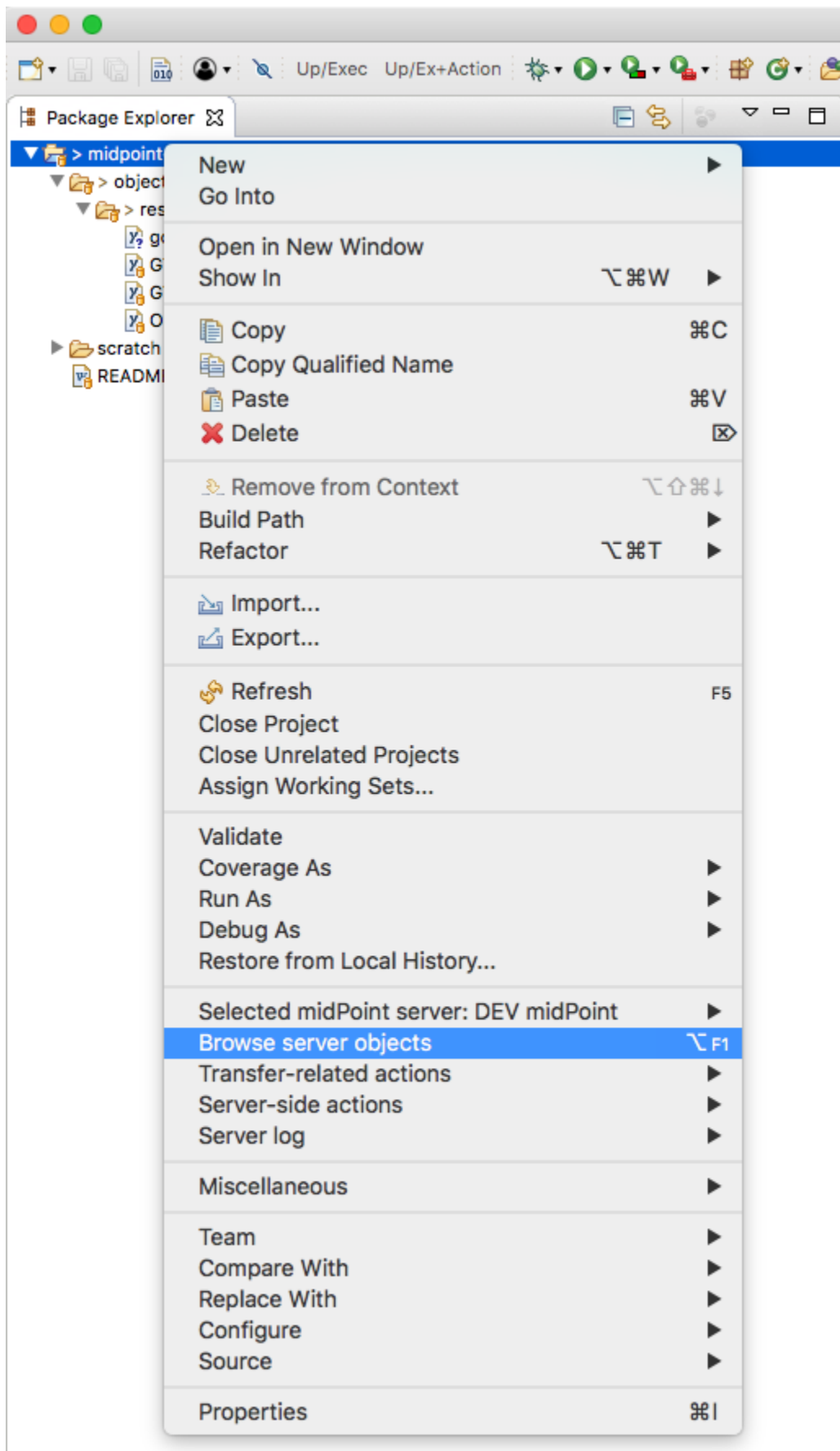
In midPoint, use the "Import object" tool to import the XML file you downloaded in step 1.



This will add a new "Google Apps" resource in the Resources list.

3. Download the resource configuration from the server

In Eclipse, right-click on the "midpoint-config-xml" project name in Package Explorer and use the midPoint plugin to "Browse server objects".



Select "Resource" in the object types list, then click Search.

Browse object on midPoint server: DEV midPoint

Please enter search criteria

Names or OIDs (one per line); or an XML query

Object types (select zero, one or more)

Organization

Report

Report output

Resource

Role

Security policy

Sequence

Service

Shadow

System configuration

Task

How to interpret object specification

☒ Names and OIDs
 ☐ Names
 ☐ OIDs
 ☐ XML query

Max # of objects to show
 Start at #

Select the "Google Apps" resource and click Download.

Browse object on midPoint server: DEV midPoint

Please enter search criteria

Names or OIDs (one per line); or an XML query

Object types (select zero, one or more)

Organization

Report

Report output

Resource

Role

Security policy

Sequence

Service

Shadow

System configuration

Task

How to interpret object specification

☒ Names and OIDs
 ☐ Names
 ☐ OIDs
 ☐ XML query

Max # of objects to show
 Start at #

Result: 5 object(s), 1 selected

| Name | Display name | Subtype | OID |
|-------------------------|--------------|---------|--------------------------------------|
| ▼ Resource: 5 object(s) | | | |
| GYBONID | | | 8d049132-b84c-4ee6-a26c-12ea0f7dd8fe |
| Google Apps | | | 2cabf644-040f-4606-84e0-9bcdcae6d83a |
| ONID LDAP DEV | | | ef2bc95b-76e0-48e2-86d6-3d4f02d3e1aa |
| ONID LDAP Testing | | | 4700b58a-e206-4664-a812-9abd48bcd06b |

Store objects in project

midpoint-config-xml

Generate

By OIDs, in one batch

N =

100

☐ Wrap created bulk actions into tasks
 ☐ Create tasks in suspended state
 ☐ Execute in raw mode
 ☐ Execute in 'dry run' mode (since midPoint 3.5)
 ☐ Use symbolic references (by name or connector type)
 ☐ Runtime resolution

4. Rename the downloaded XML file (optional)

We recommend that you rename the downloaded file to something that better reflects your organization's resources. In this example, our test Google Apps instance is called GTest, so we've renamed the configuration file to GTEST.xml.

5. Add the new resource configuration XML to the git repo

```
$ git add GTEST.xml
$ tree
midpoint-config-xml
  objects
    resources
      GTEST.xml
      GYBONID.xml
      ONIDLDAPDEV.xml
```

6. Modify the resource

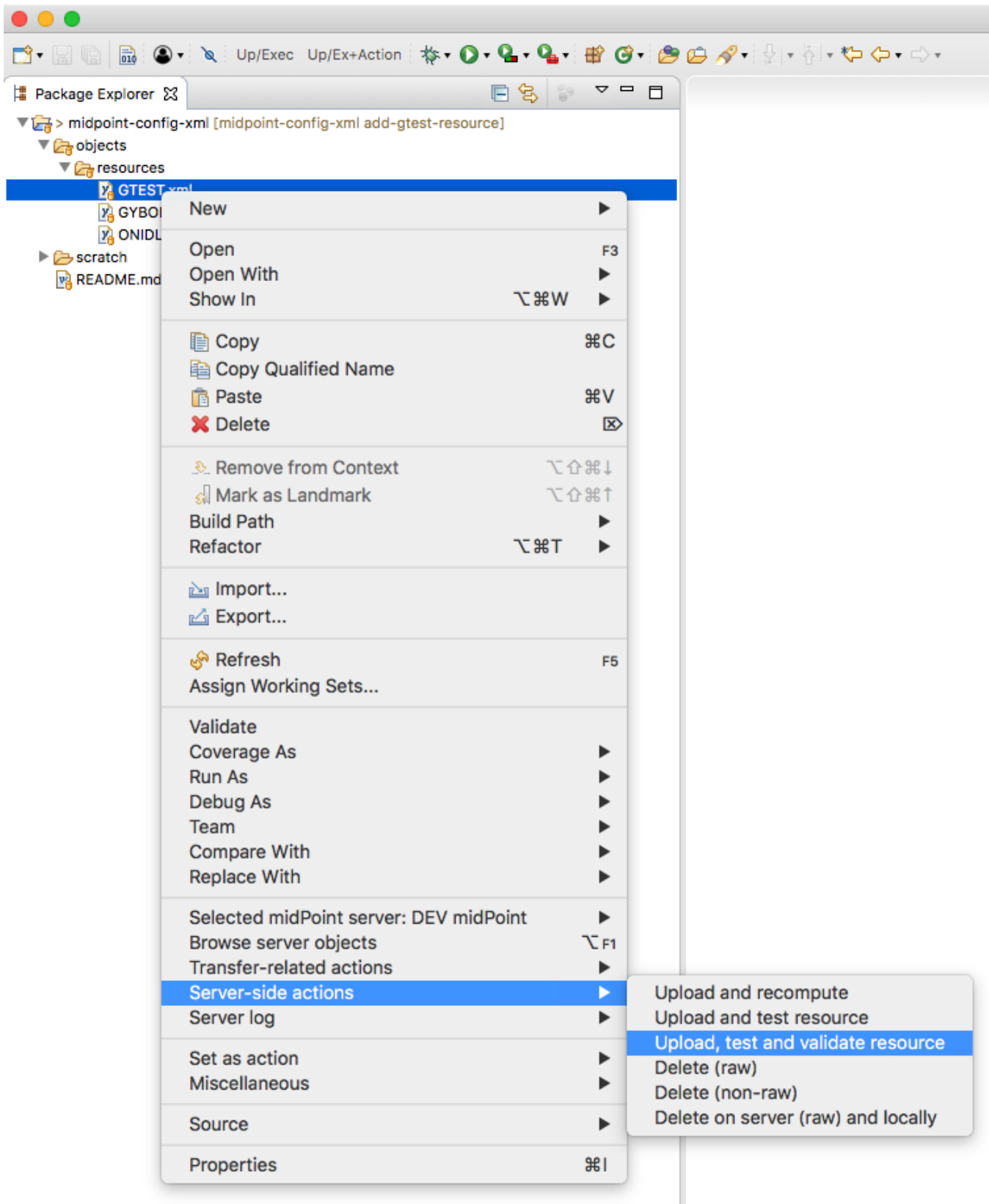
Now you can start making configuration changes to the resource XML in Eclipse. For example, change the domain to something other than the default.

```
16<connectorConfiguration xmlns:icfc="http://midpoint.evolveum.com/xml/ns/public/connector/icf-1/connector-schema-3">
17  <icfc:resultsHandlerConfiguration>
18    <icfc:enableFilteredResultsHandler>false</icfc:enableFilteredResultsHandler>
19  </icfc:resultsHandlerConfiguration>
20  <icfc:configurationProperties xmlns:gen379="http://midpoint.evolveum.com/xml/ns/public/connector/icf-1/bundle/com.ev
21    <gen379:domain>some.google.domain</gen379:domain>
22    <gen379:productId>Google-Apps</gen379:productId>
23    <gen379:skuId>Google-Apps-For-Business</gen379:skuId>
24    <gen379:autoAddLicense>false</gen379:autoAddLicense>
```

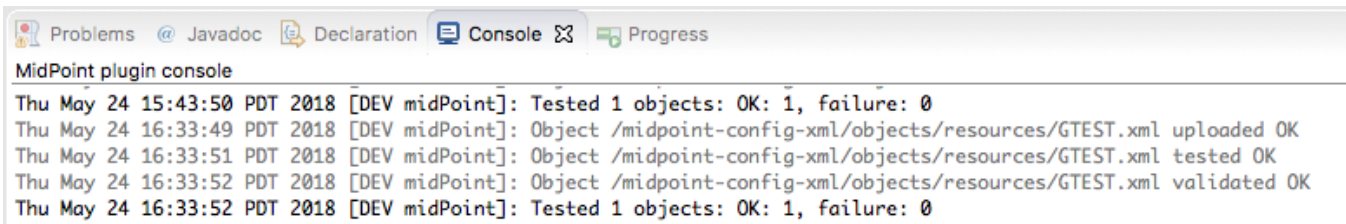
Save the file. Depending on your workflow, you may want to test your changes before committing them to git.

7. Upload the configuration change to midPoint

In Eclipse, select the modified XML file in the project explorer, right-click on it, select "Server-side actions" from the midPoint plugin menu, and then select "Upload, test, and validate".



If there are no errors, the new configuration will be applied to the resource in midPoint.



The screenshot shows a software development environment's console window. At the top, there is a tab bar with five tabs: 'Problems' (with a red bug icon), 'Javadoc' (with an '@' icon), 'Declaration' (with a document icon), 'Console' (with a speech bubble icon and is currently selected), and 'Progress' (with a green bar chart icon). Below the tabs, the title of the selected console is 'MidPoint plugin console'. The console contains five lines of log output, each starting with a timestamp and a log level in brackets. The messages describe the testing process of the MidPoint plugin, including object uploads, tests, and validations, all of which completed successfully.

```
Thu May 24 15:43:50 PDT 2018 [DEV midPoint]: Tested 1 objects: OK: 1, failure: 0
Thu May 24 16:33:49 PDT 2018 [DEV midPoint]: Object /midpoint-config-xml/objects/resources/GTEST.xml uploaded OK
Thu May 24 16:33:51 PDT 2018 [DEV midPoint]: Object /midpoint-config-xml/objects/resources/GTEST.xml tested OK
Thu May 24 16:33:52 PDT 2018 [DEV midPoint]: Object /midpoint-config-xml/objects/resources/GTEST.xml validated OK
Thu May 24 16:33:52 PDT 2018 [DEV midPoint]: Tested 1 objects: OK: 1, failure: 0
```

Stacy Brock, Oregon State University