Server Command-Line Utilities

Introduction

The Fedora server distribution comes with several useful command-line utilities. A description and usage instructions for each follows.

The scripts are located in `FEDORA_HOME/server/bin/`. In Windows, these commands resolve to batch files (.bat); in Unix, they resolve to shell scripts (.sh).

Note: There are also client command-line utilities which perform object ingest and export as well as several other functions.

This guide assumes you have correctly installed the Fedora server distribution as per the install guide, including having set up your `PATH` and `FEDORA_HOM` appropriately.

> **Information**
>
> Currently, if you are running Fedora with a servlet container other than Tomcat, these scripts will need to be manually modified for your environment to pick up the paths to the Fedora classes and required libraries from a location other than `CATALINA_HOME`.

fedora-rebuild

fedora-rebuild

Reconstitutes Fedora's indexes (the SQL database and/or Resource Index) from the FOXML and datastream files on disk.

This is an interactive utility that should be run only when the server is offline. Depending on the size of your repository, this may take minutes (thousands of objects) or hours (millions of objects) to complete.

It is useful in a variety of situations:

- *Upgrading* from a previous version of Fedora when the SQL database or Resource Index changed significantly between releases.
- *Migrating*from one SQL database product to another in an existing Fedora installation. This can be done at any time by
    1. Modifying your fedora.fcfg file to point to a properly-configured <datastore..> (see fedora.fcfg for examples)
    2. Copying the appropriate JDBC jar file into the Fedora webapp's WEB-INF/lib directory.
    3. Running a rebuild of the SQL database
- *Recovering* from inconsistencies and/or corruption of the indexes.

When you run this utility, a text menu will appear, allowing you to specify whether you need to rebuild the SQL database or the Resource Index.

To Run a Rebuild:

1. Stop the Fedora server (if using Tomcat, this can be done with the *shutdown.bat or shutdown.sh* command)
2. Run*fedora-rebuild.bat* or *fedora-rebuild.sh*
3. Select which index you want to rebuild and confirm your choice when prompted.
4. Repeat steps 2-3 to rebuild the other index, if needed.
5. Restart the Fedora server (if using Tomcat, this can be done with the *startup.bat or startup.sh*command)

> **Information**
>
> When running a SQL rebuild using MySQL with Java 1.5, it may fail with a java.lang.UnsupportedClassVersionError. This can occur if the MySQL JDBC driver you're using is a newer version. To resolve, simply run the rebuilder with Java 1.6 (ensuring your JAVA_HOME environment variable is set correctly), or use an older MySQL JDBC driver.

fedora-reload-policies

fedora-reload-policies [http|https] [username] [password]

Where:

- **http|https** - Indicates which protocol to use to send the "reload policies" signal to the running Fedora server.
- **username** - An administrative Fedora user with permission to reload polcies.
- **password** - Password for the administrative user.

Causes any new or changed repository-wide policies to take effect immediately on the running Fedora server.

As described in the document, *Fedora Authorization with XACML Policy Enforcement* , Fedora can be configured to enforce a variety of access policies. Many of these XACML policies are applied for all actions and access attempts performed on the repository as a whole. These "repository-wide" XACML policies are automatically loaded at the time the Fedora server is started. If the Fedora server administrator needs to change one or more of these repository-wide policies, this command can be used to tell the running Fedora server to reload the policies. The alternative to using this command is to stop the Fedora server and restart it.

validate-policy

validate-policy [policyFilename]

Where:

- **policyFilename** - Name of XACML file containing the new or modified policy

Schema-validates a XACML policy file.

If the Fedora server administrator creates or modifies an existing repository-wide XACML policy, the new policy **should** be run through this program to ensure that it is well-formed before attempting to install it in the Fedora server. Validating a policy in this way will ensure that it is well-formed XML and can follows the XACML XML schema.

fedora-modify-control-group

Ensure DC, RELS-EXT and RELS-INT are versionable if using Managed Content

Due to an outstanding bug FCREPO-849, if you use Managed Content for DC, RELS-EXT or RELS-INT then please make sure these datastreams are versionable (the default setting for versionable is "true", so if you haven't specified this datastream property then you are safe). Particularly take care if you are migrating an existing datastream who's VERSIONABLE property is set to "false", as this will cause problems. You will need to ensure the VERSIONABLE property is "true" before migrating.

The fedora-modify-control-group command line utility enables you to modify the control group of existing datastreams. Currently only modifying inline XML datastreams ("X") to managed content ("M") is supported. The utility can be used to modify DC, RELS-EXT and RELS-INT datastreams to managed content with the introduction of support for managed content for these datastreams in Fedora 3.4.

fedora-modify-control-group [method] [protocol] [user] [password] [pid] [dsid] [controlGroup] [addXMLHeader] [reformat] [setMIMETypeCharset]

Where:

- **method** - the operation to be performed, either *reloadpolicies* or *migratedatastreamcontrolgroup*. Use *migratedatastreamcontrolgroup* to modify the control group.
- **protocol** - the protocol to communicate with Fedora server, either http or https.
- **user** - the Fedora administrator username (e.g., fedoraAdmin).
- **password** - the password for the Fedora administrator user.
- **pid**- either
  - a single pid (eg demo:123)
  - a comma-separated list of pids (eg demo:123,demo:124)
  - the name of a file containing a list of pids (eg file:///path/to/pidfile.xml). The file may either be a simple text file containing a list of pids, or an XML file specifying pids as <pid>demo:123</pid> elements. The XML output of Fedora's basic search (the findObjects REST API method) can be used.
- **dsid** - either a single datastream identifier or a comma-separated list of datastream identifiers (eg DC or DC,RELS-EXT)
- **controlGroup** - the control group to set on the datastream. Only "M" is currently supported
- **addXMLHeader** - optional. If true, an XML declaration specifying the XML version and a character encoding of UTF-8 will be added at the start of the datastream.
- **reformat** - optional. If true, the XML will be reformatted with line breaks and indents.
- **setMIMETypeCharset** - optional. If true, a charset declaration of UTF-8 will be added to the datastream's MIMEType property if one is present

If a single pid and datastream are specified, an error will be generated if the datastream is not found. If a list of datastreams is specified, the datastreams will be upgraded only if found in the object, and no error will be given if the datastream is not found in a particular object. The output will give a full list of objects, datastreams and datastream versions migrated to the new control group.

Unable to render {include} The included page could not be found.