

# Basic Fedora Setup

This page holds notes on the current Fedora configuration, as well as misc information that must be understood when configuring Fedora.

## See also:

- [Repository Interoperability](#)
- [Fedora Resource Index](#)
- [Behavior Mechanisms](#)
- [PURL Resolution](#)
- [Fedora Batch Modify](#)
- [Coordinating Fedora Instances](#)
- [Fedora Performance Issues](#)

## Requirements

In order to run a large repository (more than 100k Fedora Objects) while using the ResourceIndex, you must have a 64-bit version of Java running on a 64-bit OS.

## Documentation

In addition to the documentation on the [Fedora site](#), Richard Green has written a very [practical tutorial](#) for new Fedora users.

## Rhyme setup

Our development machine is rhyme.dlib.indiana.edu, aka fedora-dev.dlib.indiana.edu. Its core stats are: Dual CPU 3GHz each, 6G RAM, 420G usable disk space.

Connects to Oracle database on ora-iudlu-dev (urania).

Uses port 9090 for service, 9005 for shutdown, and 9443 for redirect. This keeps it from conflicting with any Tomcat instances running on the same machine.

The fedora.sh script has been modified to increase the available Java heap space (-Xmx768m).

Since Fedora isn't very tolerant of losing its database connection, there is a cron job to stop it before the database is shutdown for backups, and another cron job to start it afterwards.

## To start Fedora on Rhyme

1. Login to your rhyme account
2. Login to the Fedora account: `su - fedora`
3. Type in password (same as the Fedora administrator password)
4. Ensure that Fedora is not running: `fedora-stop`
5. Start Fedora: `fedora-start oracle`
6. Log out

## Thalia setup

Our production machine is thalia.dlib.indiana.edu, aka fedora.dlib.indiana.edu. It has 16G of RAM and four dual core processors.

Connects to Oracle database on ora-iudlu (erato).

Uses port 9090 for service, 9005 for shutdown, and 9443 for redirect. This keeps it from conflicting with any Tomcat instances running on the same machine.

The fedora.sh script has been modified to increase the available Java heap space (-Xmx4096m). Likewise, the JAVA\_OPTS variable has been set to increase the Java heap for the "helper" Tomcat.

## Upgrading Fedora

The Fedora "migration guides" typically tell you to back up your data before upgrading. While this is a useful practice, it isn't always practical.

On rhyme, our current practice is (when the data format does not change drastically):

1. Create a new directory under /usr/local for this version
2. Install and configure the new version
3. Shut down the old version
4. Start up the new version (make sure you use the absolute path)
5. Do whatever re-indexing is needed
6. Test. If it works, update the symlink /usr/local/fedora. If it doesn't, you should be able to easily revert back to the old version.

Windows note: A similar process can be followed on Windows, but (since Windows doesn't support symlinks) it is easier to specify a FEDORA\_DEV directory, and copy the various versions of the code into there.

## Fedora 2.1 setup

We are using the default security setting **ssl-authenticate-apim**. This gives us basic SSL encryption for administrative tasks, but leaves the server open (no authentication) for basic access tasks. For this setting (and the parallel non-SSL setting) the doMediateDatastreams parameter **must** be set to false. For SSL to work correctly, the fedoraRedirectPort must be open on the machine.

When bringing up a 2.1 installation, the first time you start Fedora, the XACML policies will be initialized. These are a little over-cautious, because they don't allow objects to be created unless the request comes from the local machine. By default, Fedora configures itself to work primarily with services on the local machine. This is reflected by the IP address 127.0.0.1 in many XACML files. See the notes on the setting of fedoraServerHost in the Fedora Installation Guide. After you start the server for the first time, you should find edit all XACML files (and the beSecurity.xml file) to use the machine's public IP address. In some cases, restricting activities to the same machine is too limiting. For these cases, you can simply delete the relevant XACML file.

Steps in setting up a 2.1 server:

1. Copy the "server" distribution directory to the appropriate location (first shut down any Fedora servers or mckoi databases that are running from this location).
2. Run `fedora-setup ssl-authenticate-apim`
3. Edit the `conf/fedora.fcfg` file. (Or copy in a backed-up version)
4. Edit the `conf/beSecurity.xml` file to contain the correct IP address.
5. Make sure the database is set up correctly, and the database drivers are in Fedora's classpath (or the Tomcat common/lib directory).
6. Edit the files in `server/bin`, increasing the Java heap space settings (-Xmx). The exact settings are dependent on the machine's available memory, but should definitely be higher than the defaults.
7. If you have existing data, run `fedora-rebuild`. Rebuild the database first, then the resource index.
8. Start the server.
9. Shutdown the server. This may generate an error message, but it will work regardless.
10. Edit the XACML files to contain the correct IP address.
11. Start the server.

Notes:

- If you want to connect through SSL, make sure you use the https protocol and the redirect port (usually 9443).
- If you're starting from a blank repository and ingesting items from elsewhere, you must first ingest:
  - `bdefs`
  - `bmechs`
  - All `*util:*` objects

## Fedora 2.0 setup

Fedora 2.0 is much easier to set up than 2.1. The [official installation instructions](#) should be adequate.

However, you MUST INSTALL the patch available at [http://scripta.lib.virginia.edu/bugs/show\\_bug.cgi?id=83](http://scripta.lib.virginia.edu/bugs/show_bug.cgi?id=83) (the attachments are near the top of the page, and they download with a CGI extension that must be changed to the correct filetype)

## Current test setup (on mallow)

Must `fedora-convert-demos` to put correct hostname in demo objects.

Current McKoi username & password: `fedora`

Running on port 9090

Start with:

```
mckoi-start
fedora-start mckoi
```

Stop with:

```
fedora-stop
mckoi-stop username password
```

## Log files

The log files output by fedora only include useful information when they are set to the "finest" level, but this level creates incredibly large logs.

We will currently treat all Fedora-generated logs as disposable, being only useful for debugging. When we want to track "real" use, we will have to route everything through the Apache/Tomcat connector. Fedora 2.2 should include more organized logging output, and we may switch to that system when it is available.

## General notes

Fedora runs on its own (modified?) instance of Tomcat. It is currently not advisable to run anything besides Fedora on this version of Tomcat, because it has been tuned to give some performance enhancements for Fedora use. Be very careful when selecting ports so they don't conflict with another Tomcat that may be running on the same machine. If you change the port on which Fedora runs, it will automatically reconfigure the Fedora Tomcat, since this is really the service that's running on that port. Certain types of changes to the Tomcat config are overwritten by Fedora, so it is unlikely that we could use this copy of Tomcat for anything else.

When ingesting objects, use the admin password, not the database password.

The fedora server must be restarted for any configuration changes to take effect.

The documentation makes it seem fairly easy to move data from one repository to another: just tell the new Fedora instance to ingest all of the data from the old instance. Of course, it isn't quite this simple; as of Fedora 2.1, there are still memory leaks that prevent ingesting large numbers of objects.

Object records must be in XML form (METS or FOXML) to be ingested.

In the sample web interface, "View the Item Index" means "View the datastreams".

Fedora provides a lot of undocumented services. See the `<fedora-home>/server/tomcat/webapps/fedora/WEB-INF/web.xml` file for a full listing. The more interesting ones are:

- report: information on objects that were recently created/modified
- rsearch: search the resource index (Kowari)
- getObjectHistory/<pid>: list timestamps of changes to the object

More documentation of API-A-LITE can be found at `fedora-2.0-src/dist/userdocs/client/browser/webexp.html`

Once created, behavior definitions cannot be changed. Behavior mechanisms can only be changed marginally.

When copying Fedora objects between repositories, Fedora-level references to the local repository are changed. This means that for a datastream that redirects to another object in the same repository, or a behavior mechanism that contains the URL of the local saxon, the machine name and port number will be updated. However, references *inside* a datastream (like a reference in XSL) will not be updated.

When making a change to an XSL file, there is no simple way to reset the cache, unless the behavior mechanism explicitly uses the `clear-stylesheet-cache` option. The only thing you can do is restart Fedora (which restarts Tomcat).

## Fedora bug reporting

Bugs can be reported to [Fedora's Bugzilla](#)

user: fedora-bugreport at comm.nsd.org

pass: bugreport

## Data storage

The XML records that represent Fedora objects are stored in Fedora's objects directory (`fedora2_0_objects` by default). Underneath this directory, they are organized by a crazy date/time directory structure. Even though they don't have an XML extension, the files are really XML.

Objects that are loaded as "Fedora managed" content have their datastreams stored in the `datastreams` directory (`fedora2_0_datastreams`) using the same crazy directory structure. The file content is unchanged from the file that was loaded, but the filename is changed to reflect the PID and datastream ID.

The database coordinates all of these objects and datastreams using a fairly straightforward table setup.

If we want to convert from Managed to External content, we can just purge and re-create the datastreams. Of course, this would lose any version information.

## System limits

The Fedora project has done *some* performance testing on a repository with 1 million objects.

How many items can share a PID prefix? A PID is a 64-digit string, so if we use the prefix "iudl:", we have plenty of options for numerical data. We could even add a collection code, like "iudl:hohenberger-1214".

While it is difficult to determine exactly what the real use will be, we have tested the Fedora-based Slocum Puzzles webapp with 50 simultaneous users making continuous requests. The server slowed down, but was still giving response pages within a reasonable amount of time (<5 seconds). The bottleneck seemed to be the speed at which the `purResolver` app could serve images out of Fedora. With improvements to this system (possibly copying the thumbnails to a static location), we should be able to increase the performance.

## NSDL setup

(from [Representing Contextualized Information in the NSDL](#))

The NDR has over 2.1 million digital objects - 882,000 of them matching metadata from the MR, 1.2 million of them representing NSDL resources, and several hundred representing other information objects - agents, services, etc., - in the NDR data model. The representation of the relationships among these objects (those defined by the NDR data model and those internal to the Fedora digital object representation) produces over 165 million RDF triples in the triple-store. We have found that ingest into the NDR takes about .7 seconds per object - making data load for this rich information environment a non-trivial task.

The platform for our NDR production environment is a Dell 6850 server with dual 3Ghz Xeon processors, 32Gb of 400Mhz memory and 517Gb of SCSI RAID disk with 80MB/second sustained performance. This server is running 64-bit LINUX, for reasons outlined later. We note that the 2006 cost for this production server is about 22K USD.

The Kowari-based resource index requires over 54 GB of virtual memory.

## Purging a repository

(from Nikolai Schwertner, via the Fedora mailing list)

The best way to purge all objects from a Fedora repository is to reset the repository. Here are the steps:

1. Stop the Fedora instance.
2. Drop the Fedora database, and create a blank Fedora database with the same permission/privileges OR empty the tables using an external SQL tool
3. Delete the files and subdirectories from the Fedora objects, datastreams, temp, and resourceIndex directories.
4. Start the Fedora server.