

# Deploying the Bamboo CMIS-Fedora Server

- [Overview](#)
- [Fedora Configuration](#)
- [J2EE Container](#)
- [App Configuration](#)
  - [CMIS Authentication/Authorization Notes](#)

## Overview

The Bamboo CMIS-Fedora web application—henceforth, the App—is a Java web application that extends classes in the Apache Chemistry Java implementation of CMIS in order to support Bamboo's extension of CMIS document and folder types, and to use Fedora Commons Repository for robust persistence in place of the Chemistry's File Share Repository.

While the App is designed to run independently of the Fedora instance it uses for persistence, it does require a running Fedora instance. These instructions for deploying an instance of the Bamboo CMIS-Fedora Server assume that Fedora have been installed and is running. See <http://www.fedora-commons.org/>.

These instructions also assume that you have installed a J2EE container such Apache Tomcat, Jetty, or JBoss and that you are familiar with how WARs are deployed in your container.

## Fedora Configuration

The App assumes that Fedora has been configured with Resource Indexing turned on (property "level"=1) and the Resource Index "syncUpdates" property set to true.

By default, syncUpdate is set to false in order to "significantly reduce roundtrip times for API-M operations" (the fedora.fcfg configuration property comment). However, Bamboo CMIS objects are inherently hierarchical, meaning that we must declare the parent of an underlying CMIS folder or document when creating the object. We've found when creating a hierarchy of objects, e.g. a book with child pages, that unless the index is updated with each object insertion, as is guaranteed with syncUpdate set to true, insertion of a child object can fail because its parent object is not found in the Resource Index, owing to the backlog of updates to the index.

## J2EE Container

Fedora runs in its own J2EE container. We've found that it is possible to add the CMIS-Fedora Server WAR to Fedora's container, but we note that the load order for WARs cannot be controlled in Tomcat, and so it is possible that the CMIS-Fedora Server could be available before Fedora has come online. So long as the CMIS-Fedora Server is not serving requests until Fedora is available, all is well.

## App Configuration

The App is configured prior to building the WAR that will be deployed in your container. The source code is at `$(REPOSITORY_ROOT)/work-space-repository/fedora_cmis_repository/trunk/fedora-cmis-repository/` – at the following URL on Sourceforge: [http://svn.code.sf.net/p/projectbamboo/code/work-space-repository/fedora\\_cmis\\_repository/trunk/fedora-cmis-repository/](http://svn.code.sf.net/p/projectbamboo/code/work-space-repository/fedora_cmis_repository/trunk/fedora-cmis-repository/) (This is a NetBeans project; NetBeans has been used to execute the WAR build.) The configuration items are set in the `/WEB-INF/classes/repository.properties` file in the App's source tree. Several properties need to be set.

```
type=fedora
fedoraUrl=http://localhost:8080/fedora
fedoraUsername=fedoraAdmin
fedoraPassword=password
```

The type property controls whether we are using Fedora or the Chemistry File Share Repository.

We have made the naive assumption that the fedoraUsername will be "fedoraAdmin". (We have not assumed that the fedoraPassword will be "password") All objects within the CMIS hierarchy are owned by fedoraAdmin from Fedora's point of view. This assumption is baked into the App's code (SimpleFedoraRepository.java). (An obvious refactoring would be to assign the Fedora user based on the repository.properties setting.) From CMIS's point of view, however, object ownership and permissions are managed through BambooACLs; object owners are principals managed by the CMIS clients.

The fedoraUrl is just the URL to the Fedora instance you are using.

## CMIS Authentication/Authorization Notes

We have also made a naive assumption about how CMIS clients authenticate to the CMIS-Fedora Server; that is, we have accepted the out-of-the box Chemistry Basic Authorization (username/password) approach to authentication and their simple "readwrite vs readonly" approach baseline authorization. As noted elsewhere, we have extended `org.apache.chemistry.opencmis.server.shared.BasicAuthCallContextHandler` in `org.projectbamboo.ci.BambooContextHandler` to gather two attributes from the HTTP request headers: "X-Forwarded-User" and "X-CMIS-Principals". These attributes are used in conjunction with BambooACLs set at the object level to manage finer grained authorization and object ownership declarations.

You should look to Apache Chemistry to for ways of improving authentication. See <http://chemistry.apache.org/java/developing/client/dev-client-bindings.html#OpenCMISClientBindings-CustomAuthenticationProvider>