

# Jackrabbit 2 in Sling

## Using Jackrabbit Version 2.0

- [Update 7. Feb. 2011](#)
- [Introduction](#)
- [Changes](#)
  - [jcr/base](#)
  - [jcr/jackrabbit-server](#)
  - [jcr/webdav](#)
  - [jcr/jackrabbit-usermanager](#)
  - [jcr/jackrabbit-accessmanager](#)
  - [contrib/jcr/jackrabbit-api](#)
  - [launchpad/bundles](#)
- [Requirements](#)
  - [Tika Extension](#)

### Update 7. Feb. 2011

Sling has in the meantime been upgraded to using Jackrabbit 2.1.1 and using the Tika and Derby libraries as bundles. So the contents of this page now is of historic value only.

### Introduction

The upcoming Jackrabbit 2.0 release will be the first Jackrabbit release supporting the recently publish [JCR 2.0](#) specification. With support for this new release, currently available from the central maven repository as the third beta release (2.0-beta3), I also propose to split the embedded Jackrabbit Repository module and take out two important helper libraries:

- The Derby core library (providing the embedded Derby database used as the default persistence layer by Jackrabbit 2.0) is available as a bundle and thus the embedded Jackrabbit Repository bundle will have an optional import of the Derby JDBC driver package (nothing more is needed, since Jackrabbit accesses Derby using plain JDBC).
- Starting with Jackrabbit 2.0 the built-in indexer uses the Apache Tika library to parse and index content. Since the raw Tika libraries are available as bundles (and TIKAXXXX proposes a convenience full Tika bundle) the dependencies to the Tika packages are imported.

This dramatically reduces the size of the Jackrabbit bundle.

The following libraries will (for now) remain in the embedded Jackrabbit Repository bundle:

- Xerces – traditionally it has been very hard to bundleize the XML based libraries. Also version 2.8.1 of the `xercesImpl` library used by Jackrabbit is not available as a bundle. Thus it is easiest to keep this library in the bundle.
- Lucene Core – Same as Xerces, the lucene core library is not available as a bundle currently (not even the most recent 3.0 version). Thus we keep this library inside Jackrabbit, too.
- Helpers – further support libraries from the Jackrabbit project itself and other projects are kept with the embedded Jackrabbit Repository bundle: `jackrabbit-jcr-commons`, `jackrabbit-spi-commons`, `jackrabbit-spi`, `concurrent`, `jackrabbit-jcr-rmi`.

Further investigation will have to show to what extent the SPI and SPI Commons libraries can and should be used as bundles.

**NOTE** We do **not** upgrade the default JCR API import defined in the parent POM to version 2.0 to ensure that by default Sling is still able to run on plain JCR 1.0 repositories. Those parts of Sling really requiring JCR 2.0 functionality will have to explicitly set the required JCR API version to 2.0.

Progress in this upgrade is tracked in [SLING-1212](#). Prototype implementation can be found in my whiteboard at <https://svn.apache.org/repos/asf/sling/whiteboard/fmeschbe/jackrabbit2upgrade>

### Changes

#### **jcr/base**

The `jcr/base` bundle contains a utility class providing access to the Jackrabbit Access control implementation. This utility class has to be updated to use the new JCR 2.0 API instead of the transient Jackrabbit API previously used.

In addition we finally remove the SessionPool support we have disabled for a quite some time now. Though in the early days of Jackrabbit session setup was quite expensive, nowadays this setup is fairly quick and a session pool poses more issues – mainly for cleaning up and reusing sessions – than it solves problems.

#### **jcr/jackrabbit-server**

The biggest changes are concerned with the embedded Jackrabbit Repository bundle. As described above, one change is the exclusion of the indexing (parsing) and derby libraries from the bundle itself. This reduces the size of this bundle considerably. In addition all Jackrabbit dependencies are upgraded to the 2.0-beta3 version.

Finally the security plugin support classes are converted to the new JCR 2.0 API. The pluggable `DefaultLoginModule` and `DefaultAccessManager` classes still remain to provide dynamic extensibility based on the OSGi service registry.

## **jcr/webdav**

The upgrade of the WebDAV provider requires integration of Tika API because starting with Jackrabbit 2.0 Tika is also used for MIME type resolution in the Jackrabbit project. For this reason, the Sling WebDAV bundle will convert the existing wrapper to a Tika `Detector` baking decisions – as before – on the Sling `MimeTypeService`.

The Sling WebDAV bundle will import the Tika API as does the `jcr/jackrabbit-server` bundle.

## **jcr/jackrabbit-usermanager**

The Jackrabbit User Manager bundle must of course be upgraded.

TBD

## **jcr/jackrabbit-accessmanager**

The Jackrabbit Access Manager bundle must of course be upgraded.

TBD

## **contrib/jcr/jackrabbit-api**

This project is obsolete because the Jackrabbit project provides the Jackrabbit API library as a bundle for quite some time now. Thus this project can safely be removed.

## **launchpad/bundles**

The `launchpad/bundles` project is modified to include the required JCR and Jackrabbit bundles:

- `javax.jcr:jcr:2.0` – The JCR 2.0 library comes ready-set as a bundle and directly be deployed
- `org.apache.jackrabbit:jackrabbit-api:2.0-beta3` – The Jackrabbit API must be upgraded to the correct version
- `org.apache.jackrabbit:jackrabbit-jcr-commons:2.0-beta3` – Likewise the Jackrabbit JCR Commons library is to be upgraded
- `org.apache.derby:derby:10.5.3.0_1` – imported by the embedded Jackrabbit Repository bundle
- `org.apache.tika:tika-full:0.6-SNAPSHOT` – imported by the embedded Jackrabbit Repository bundle and the WebDAV support bundle (requires manual build)
- `commons-fileupload:commons-fileupload:1.2.1` – imported by the WebDAV support bundle

Besides that the current SNAPSHOTs of the Jackrabbit embedding bundles must of course be included.

## **Requirements**

### **Tika Extension**

To be able to simply import the Apache Tika packages a convenience full bundle of the Tika Core and Parser libraries as well as some of the core support libraries would be very helpful. As such I prepared such a bundling and created [TIKA-340](#) to propose this addition to the Tika project. To build and use the Jackrabbit 2.0 support yourself checkout the Tika project, apply the patch and install (at least) the `tika-full` bundle in your local maven repository.