

Launching a JCR Repository inside Equinox

Launching a JCR Repository inside Equinox

Today I was contacted with a request to setup the `jcr/jackrabbit-server` bundle in an OSGi framework such that a JCR Repository is being started and can be accessed from within the OSGi framework and over RMI from outside of the framework. It happens to be that the framework of choice has been Equinox.

So, to better support I decided to setup my own Equinox framework and install required bundles.

Getting the Equinox Framework

First of all, you need the Equinox OSGi framework. You can grab this from the Equinox download page at <http://download.eclipse.org/eclipse/equinox/>. I decided to use the latest release stuff, which at the time of this writing is 3.3.2. So I downloaded the framework jar file [org.eclipse.osgi_3.3.2.R33x_v20080105.jar](#)

Once this has been downloaded, it can easily be started from the command line. As I required the console prompt, I added the `-console` command line option:

```
$ java -jar org.eclipse.osgi_3.3.2.R33x_v20080105.jar -console

osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105

osgi>
```

The console gives you the `osgi>` prompt to which I entered a first `ss` command to get a short status of installed bundles. This lists just the system bundle with ID 0. This console is used by us to interactively install and start additional bundles, until we get a running Jackrabbit Repository.

To get a list of available commands just enter the `help` command. Some commands which we will use later on are :

- **ss** – display a short status list of installed bundles
- **status** – display a long status list of installed bundles
- **bundle <id>** – display detailed information on the bundle with the given `<id>`. This list includes registered and used services, import and export packages and some more information.
- **install <url>** – installs a bundle from the given `<url>`. Note that this must be a real URL, that is to install a file from the local filesystem, the `file:` protocol must be specified.
- **start <id>** – starts the bundle with the given `<id>`.
- **stop <id>** – stops the bundle with the given `<id>`.
- **uninstall <id>** – uninstalls the bundle with the given `<id>`.

OSGi Compendium

First we need to install OSGi Compendium Services bundles:

1. The Compendium Services API bundle
2. A LogService implementation
3. A Configuration Admin Service implementation
4. A Declarative Services implementation

The Compendium Services API bundle

We chose to mainly use Equinox implementations, so we get the [org.eclipse.osgi.services_3.1.200.v20070605.jar](#) bundle to install the OSGi Compendium Services API:

```

osgi> install <url>
Bundle id is 1

osgi> start 1

osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE    org.eclipse.osgi.services_3.1.200.v20070605

osgi>

```

Note that bundles are not automatically started after the installation. Therefore we have to manually start the bundles after the installation. This manual start is only required after the installation. On framework restarts all bundles, which were started when the framework was last shutdown, are automatically restarted.

A LogService implementation

We continue with the LogService. Here we could of course also choose to use the Equinox implementation. But as I know, that Jackrabbit uses [SLF4J](#) for logging and some other components use log4j and Jakarta Commons Logging, I strongly suggest to use the LogService implementation from the Sling project. This bundle unifies all logging requests from the OSGi LogService and the SLF4J, log4j and Jakarta Commons Logging API into a single log channel. This makes problem traking much easier.

The Sling LogService implementation is available from the Apache Snapshot repository and may be directly installed into the OSGi framework:

```

osgi> install <url>
Bundle id is 2

osgi> start 2
30.04.2008 21:13:24.642 *INFO* [OSGi Console] org.apache.sling.osgi.log.LogServiceFactory LogManager: Logging
set up from context
30.04.2008 21:13:24.645 *INFO* [OSGi Console] org.apache.sling.osgi.log Service [org.apache.sling.osgi.log.
LogServiceFactory,22] ServiceEvent REGISTERED
30.04.2008 21:13:24.646 *INFO* [OSGi Console] org.apache.sling.osgi.log Service [org.apache.sling.osgi.log.
LogReaderServiceFactory,23] ServiceEvent REGISTERED

osgi> 30.04.2008 21:13:24.646 *INFO* [Framework Event Dispatcher] org.apache.sling.osgi.log BundleEvent STARTED

osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE    org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE    org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT

osgi>

```

We installed the LogService implementation early in the process to be able to get some logging in the future processing. You may have noticed, that LogService already kicked in writing logs to stdout. In the further steps you will notice, that each bundle installation and start will be accompanied with log messages.

A Configuration Admin Service implementation

Core configuration to launche the Jackrabbit repository is maintained using the OSGi Configuration Admin Service. For the moment this configuration just encompasses the filesystem location of the repository and the path to the repository configuration file. In the future this may be extended if configuration options for Jackrabbit may change.

For the sake of using Equinox bundles, lets take the Equinox implementation of the Configuration Admin Service available from [org.eclipse.equinox.cm_3.2.0.v20070116.jar](#).

```

osgi> install <url>
Bundle id is 4

osgi> 30.04.2008 21:22:16.042 *INFO* [Framework Event Dispatcher] org.eclipse.equinox.cm BundleEvent INSTALLED
start 4
30.04.2008 21:22:19.921 *INFO* [Framework Event Dispatcher] org.eclipse.equinox.cm BundleEvent RESOLVED
30.04.2008 21:22:19.949 *INFO* [OSGi Console] org.eclipse.equinox.cm Service [24] ServiceEvent REGISTERED

30.04.2008 21:22:19.973 *INFO* [Framework Event Dispatcher] org.eclipse.equinox.cm BundleEvent STARTED
osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE     org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE     org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE     org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT
4       ACTIVE     org.eclipse.equinox.cm_3.2.0.v20070116

osgi>

```

A Declarative Services implementation

The last OSGi Compendium Service implementation we need is Declarative Services. I of course also tried the Equinox implementation of this specification. But unfortunately the Sling Jackrabbit Repository bundle cannot be used with this implementation. This is of course not a good thing and we will try to find out, what is wrong here. This issue is tracked as [SLING-408](#). So we take the Apache Felix implementation from [org.apache.felix.scr-1.0.0.jar](#).

```

osgi> install <url>
Bundle id is 5
30.04.2008 22:12:34.988 *INFO* [Framework Event Dispatcher] org.apache.felix.scr BundleEvent INSTALLED

osgi> start 5
30.04.2008 22:12:38.032 *INFO* [Framework Event Dispatcher] org.apache.felix.scr BundleEvent RESOLVED
30.04.2008 22:12:38.033 *INFO* [OSGi Console] org.apache.felix.scr Service [Declarative Services Management
Agent,25] ServiceEvent REGISTERED

osgi> 30.04.2008 22:12:38.039 *INFO* [Framework Event Dispatcher] org.apache.felix.scr BundleEvent STARTED
ss

Framework is launched.

id      State      Bundle
0       ACTIVE     org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE     org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE     org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT
4       ACTIVE     org.eclipse.equinox.cm_3.2.0.v20070116
5       ACTIVE     org.apache.felix.scr_1.0.0

osgi>

```

Jackrabbit Bundles

The next (and final) big step is to install and start the bundles required to embed and start the Jackrabbit repository. In short, we are going to install the following bundles:

1. **Jackrabbit Commons and Jakarta Commons Collections Bundles** – Library bundles by the rest of the Jackrabbit bundles.
2. **Sling JCR API and Base Bundles** – These bundles export the JCR API and the base implementation for session pooling on which the embedded Jackrabbit Repository of Sling is based.
3. **Jackrabbit Server and Text Extractors Bundle** – The actual embedded Jackrabbit repository bundle and a library used for extended indexing and query support.

Commons Collections and Jackrabbit Commons

We start with the helper libraries Jakarta Commons Collections and Jackrabbit Commons. Fortunately the latest releases of both libraries already come built ready as OSGi bundles. So we just grab the respective libraries from the Maven 2 repository and install them as is:

```
osgi> install <url>
Bundle id is 6

osgi> 01.05.2008 20:31:33.553 *INFO* [Framework Event Dispatcher] org.apache.commons.collections BundleEvent
INSTALLED
start 6

osgi> 01.05.2008 20:31:38.374 *INFO* [Framework Event Dispatcher] org.apache.commons.collections BundleEvent
RESOLVED
01.05.2008 20:31:38.374 *INFO* [Framework Event Dispatcher] org.apache.commons.collections BundleEvent STARTED
install <url>
Bundle id is 7
01.05.2008 20:32:58.959 *INFO* [Framework Event Dispatcher] org.apache.jackrabbit.jackrabbit-jcr-commons
BundleEvent INSTALLED

osgi> start 7
org.osgi.framework.BundleException: The bundle could not be resolved. Reason: Missing Constraint: Import-
Package: javax.jcr; version="0.0.0"
    at org.eclipse.osgi.framework.internal.core.BundleHost.startWorker(BundleHost.java:305)
    at org.eclipse.osgi.framework.internal.core.AbstractBundle.start(AbstractBundle.java:260)
    at org.eclipse.osgi.framework.internal.core.AbstractBundle.start(AbstractBundle.java:252)
    at org.eclipse.osgi.framework.internal.core.FrameworkCommandProvider._start(FrameworkCommandProvider.
java:260)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
    at org.eclipse.osgi.framework.internal.core.FrameworkCommandInterpreter.execute
(FrameworkCommandInterpreter.java:150)
    at org.eclipse.osgi.framework.internal.core.FrameworkConsole.docommand(FrameworkConsole.java:300)
    at org.eclipse.osgi.framework.internal.core.FrameworkConsole.console(FrameworkConsole.java:285)
    at org.eclipse.osgi.framework.internal.core.FrameworkConsole.run(FrameworkConsole.java:221)
    at java.lang.Thread.run(Thread.java:619)

osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE    org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE    org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT
4       ACTIVE    org.eclipse.equinox.cm_3.2.0.v20070116
5       ACTIVE    org.apache.felix.scr_1.0.0
6       ACTIVE    org.apache.commons.collections_3.2.1
7       INSTALLED org.apache.jackrabbit.jackrabbit-jcr-commons_1.4.2

osgi>
```

Oops ! Bundle 7, the Jackrabbit Commons library bundle, fails to start. The reason for this is, that this library depends on the Content Repository for Java (JCR) API classes, which have not been installed yet and which are not available from the environment either.

This is not actually a problem because the Sling JCR API bundle exports just these packages. So let's just continue with the next steps.

Sling JCR API and Base Bundles

The Sling JCR API and Base Bundles can be installed directly from the Apache Snapshot Repository, which we just do now:

```

osgi> install <url>
Bundle id is 8

osgi> 01.05.2008 20:37:21.866 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.api BundleEvent INSTALLED
start 8
01.05.2008 20:37:28.099 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.api BundleEvent RESOLVED
01.05.2008 20:37:28.100 *INFO* [Framework Event Dispatcher] org.apache.jackrabbit.jackrabbit-jcr-commons
BundleEvent RESOLVED

osgi> 01.05.2008 20:37:28.100 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.api BundleEvent STARTED
install <url>
Bundle id is 9
01.05.2008 20:38:45.563 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.base BundleEvent INSTALLED

osgi> start 9
01.05.2008 20:38:51.069 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.base BundleEvent RESOLVED

01.05.2008 20:38:51.069 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.base BundleEvent STARTED
osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE    org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE    org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT
4       ACTIVE    org.eclipse.equinox.cm_3.2.0.v20070116
5       ACTIVE    org.apache.felix.scr_1.0.0
6       ACTIVE    org.apache.commons.collections_3.2.1
7       RESOLVED  org.apache.jackrabbit.jackrabbit-jcr-commons_1.4.2
8       ACTIVE    org.apache.sling.jcr.api_2.0.0.incubator-SNAPSHOT
9       ACTIVE    org.apache.sling.jcr.base_2.0.0.incubator-SNAPSHOT

osgi>

```

Now that we have installed the API and base bundles, we see that bundle 7 is now in the *RESOLVED* state. That is, the imports could now be completely resolved. In fact, the bundle has been resolved because the JCR Base Bundle (Bundle 9) requires the classes exported by bundle 7. So we can now start bundle 7:

```

osgi> start 7

osgi> 01.05.2008 20:39:03.845 *INFO* [Framework Event Dispatcher] org.apache.jackrabbit.jackrabbit-jcr-commons
BundleEvent STARTED
ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE    org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE    org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT
4       ACTIVE    org.eclipse.equinox.cm_3.2.0.v20070116
5       ACTIVE    org.apache.felix.scr_1.0.0
6       ACTIVE    org.apache.commons.collections_3.2.1
7       ACTIVE    org.apache.jackrabbit.jackrabbit-jcr-commons_1.4.2
8       ACTIVE    org.apache.sling.jcr.api_2.0.0.incubator-SNAPSHOT
9       ACTIVE    org.apache.sling.jcr.base_2.0.0.incubator-SNAPSHOT

osgi>

```

Jackrabbit Repository Bundles

Finally we install and start the Jackrabbit Repository and Text Extractor Bundles:

```
osgi> install <url>
01.05.2008 20:45:57.183 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.jackrabbit-text-extractors
BundleEvent INSTALLED
Bundle id is 10

osgi> install <url>
Bundle id is 11

osgi> 01.05.2008 21:19:05.079 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.jackrabbit.server
BundleEvent INSTALLED
start 10

osgi> 01.05.2008 21:19:27.219 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.jackrabbit-text-
extractors BundleEvent STARTED
start 11
01.05.2008 21:19:30.940 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.jackrabbit.server BundleEvent
RESOLVED
01.05.2008 21:19:31.058 *INFO* [OSGi Console] org.apache.sling.jcr.jackrabbit.server.Activator Creating default
config for Jackrabbit in jackrabbit
01.05.2008 21:19:31.075 *INFO* [OSGi Console] org.apache.sling.jcr.jackrabbit.server.Activator
verifyConfiguration: Created configuration org.apache.sling.jcr.jackrabbit.server.SlingServerRepository-
1209669571062-0 for org.apache.sling.jcr.jackrabbit.server.SlingServerRepository
01.05.2008 21:19:31.096 *INFO* [OSGi Console] org.apache.sling.jcr.jackrabbit.server Service [org.apache.sling.
jcr.jackrabbit.server.JndiRegistrationSupport,26] ServiceEvent REGISTERED
01.05.2008 21:19:31.153 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Using JNDI context
{java.naming.provider.url=http://incubator.apache.org/sling, java.naming.factory.initial=org.apache.jackrabbit.
core.jndi.provider.DummyInitialContextFactory} to register repositories
01.05.2008 21:19:31.153 *INFO* [OSGi Console] org.apache.sling.jcr.jackrabbit.server Service [org.apache.sling.
jcr.jackrabbit.server.RmiRegistrationSupport,27] ServiceEvent REGISTERED
01.05.2008 21:19:31.155 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Using RMI Registry
port 1099
01.05.2008 21:19:31.156 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Service [org.apache.
sling.jcr.jackrabbit.server.SlingServerRepository,28] ServiceEvent REGISTERED

osgi> 01.05.2008 21:19:31.169 *INFO* [Framework Event Dispatcher] org.apache.sling.jcr.jackrabbit.server
BundleEvent STARTED
01.05.2008 21:19:31.283 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.RepositoryImpl Starting
repository...
01.05.2008 21:19:31.294 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/repository
01.05.2008 21:19:31.433 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.nodetype.NodeTypeRegistry no
custom node type definitions found
01.05.2008 21:19:31.453 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/version
01.05.2008 21:19:32.555 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.persistence.db.
DatabasePersistenceManager Database: Apache Derby / 10.2.1.6 - (452058)
01.05.2008 21:19:32.555 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.persistence.db.
DatabasePersistenceManager Driver: Apache Derby Embedded JDBC Driver / 10.2.1.6 - (452058)
01.05.2008 21:19:33.060 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/version/blobs
01.05.2008 21:19:33.105 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.RepositoryImpl initializing
workspace 'default'...
01.05.2008 21:19:33.106 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/workspaces/default
01.05.2008 21:19:33.620 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.persistence.db.
DatabasePersistenceManager Database: Apache Derby / 10.2.1.6 - (452058)
01.05.2008 21:19:33.620 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.persistence.db.
DatabasePersistenceManager Driver: Apache Derby Embedded JDBC Driver / 10.2.1.6 - (452058)
01.05.2008 21:19:33.870 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/workspaces/default/blobs
01.05.2008 21:19:33.914 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.RepositoryImpl workspace
'default' initialized
01.05.2008 21:19:34.043 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/repository/index
01.05.2008 21:19:34.865 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.query.lucene.SearchIndex Index
initialized: jackrabbit/repository/index Version: 2
01.05.2008 21:19:34.868 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.fs.local.LocalFileSystem
LocalFileSystem initialized at path jackrabbit/workspaces/default/index
01.05.2008 21:19:34.869 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.query.lucene.SearchIndex Index
initialized: jackrabbit/workspaces/default/index Version: 2
01.05.2008 21:19:34.870 *INFO* [SCR Component Actor] org.apache.jackrabbit.core.RepositoryImpl Repository
```

```

started
01.05.2008 21:19:35.017 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Service [org.apache.sling.jcr.jackrabbit.server.SlingServerRepository-1209669571062-0,29] ServiceEvent REGISTERED
01.05.2008 21:19:35.018 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Repository bound to JNDI as jackrabbit
01.05.2008 21:19:35.164 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Using private RMI Registry at 1099
01.05.2008 21:19:35.165 *INFO* [SCR Component Actor] org.apache.sling.jcr.jackrabbit.server Repository bound to //bslm-046.corp.day.com:1099/jackrabbit
ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.3.2.R33x_v20080105
1       ACTIVE    org.eclipse.osgi.services_3.1.200.v20070605
2       ACTIVE    org.apache.sling.osgi.log_2.0.0.incubator-SNAPSHOT
4       ACTIVE    org.eclipse.equinox.cm_3.2.0.v20070116
5       ACTIVE    org.apache.felix.scr_1.0.0
6       ACTIVE    org.apache.commons.collections_3.2.1
7       ACTIVE    org.apache.jackrabbit.jackrabbit-jcr-commons_1.4.2
8       ACTIVE    org.apache.sling.jcr.api_2.0.0.incubator-SNAPSHOT
9       ACTIVE    org.apache.sling.jcr.base_2.0.0.incubator-SNAPSHOT
10      ACTIVE    org.apache.sling.jcr.jackrabbit-text-extractors_1.4.0.incubator-SNAPSHOT
11      ACTIVE    org.apache.sling.jcr.jackrabbit.server_2.0.0.incubator-SNAPSHOT

osgi>

```

Wow! After starting the Text Extractors and Jackrabbit Server bundles, the repository is immediately starting and is then available as an OSGi service of type `javax.jcr.Repository` and over RMI using the RMI URL `//localhost:1099/jackrabbit`

That's it. Now, you can install more bundles and enjoy the fun of programming the Content Repository for Java API.

Links

This is a complete list of the links to the files we needed during download. Note, that some links are behind a mirroring scripts (mostly the Equinox download links).

Description	Link
Equinox OSGi Framework	org.eclipse.osgi_3.3.2.R33x_v20080105.jar
OSGi Compendium Services API	org.eclipse.osgi.services_3.1.200.v20070605.jar
Sling OSGi LogService Implementation	org.apache.sling.osgi.log-2.0.0-incubator-20080430.113337-26.jar
Equinox OSGi Configuration Admin Service Implementation	org.eclipse.equinox.cm_3.2.0.v20070116.jar
Felix OSGi Declarative Service Implementation	org.apache.felix.scr-1.0.0.jar
Jakarta Commons Collections	commons-collections-3.2.1.jar
Jackrabbit Commons	jackrabbit-jcr-commons-1.4.2.jar
Sling JCR API Bundle	org.apache.sling.jcr.api-2.0.0-incubator-20080430.113337-29.jar
Sling JCR Base Bundle	org.apache.sling.jcr.base-2.0.0-incubator-20080501.182241-31.jar
Sling Jackrabbit Text Extractors Bundle	org.apache.sling.jcr.jackrabbit-text-extractors-1.4.0-0003-incubator-20080430.113337-9.jar
Sling Jackrabbit Embedded Repository Bundle	org.apache.sling.jcr.jackrabbit.server-2.0.0-incubator-20080501.191424-29.jar