

Module Reorganization Proposal

IMPLEMENTED

Status: IMPLEMENTED

In issue SLING-83 Bertrand proposes to improve our project layout to make it easier to developers to find their way through the projects making up Sling. On this page, I would like to propose a new structure for further discussion.

[\[Naming Conventions \]](#) [\[Top Level Projects \]](#) [\[Sling Commons Projects \]](#) [\[JCR Related Projects \]](#) [\[Sling Launcher Projects \]](#) [\[Maven Plugin Projects \]](#) [\[microscaling Projects \]](#) [\[OSGi Projects \]](#) [\[Scripting Engine Projects \]](#) [\[Sling Main Projects \]](#) [\[Removed and Replaced Projects \]](#)

Naming Conventions

Projects

- *groupId*: All projects have the same *groupId* `org.apache.sling`
- *artifactId*: The *artifactId* is fully qualified with the *groupId* plus a name built from the project folder as listed below with slashes / and dashes - replaced by dots .. See below for exceptions.

The final artifact name - `pom.build.finalName` - is the Maven 2 default consisting of the *artifactId* and the *version*. The `Bundle-SymbolicName` is defined in the parent pom as being just the *artifactId*.

Exceptions to *artifactId* rule

- The parent project's *artifactId* is `sling`
- The *artifactId* of the projects in the `sling` folder is just the project name without the parent folder `sling`, e.g `core` for the Sling core project.
- The Maven 2 plugin projects in the `maven` folder follow the naming convention for Maven 2 plugins.

Java Packages

Each project consists of a series of Java Packages. To prevent collision between projects with respect to Java Packages, each project has a root Java Package which is the fully qualified Maven 2 project name consisting of the *artifactId* such that the `Bundle symbolic name` and the root Java Package are the same.

Top Level Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|----------|--------|------------------------|----------------|
| OK | fmeschbe | parent | org.apache.sling.sling | Parent Project |
| OK | fmeschbe | api | org.apache.sling.api | Core API |

The Sling parent project and the Sling API projects are the only remaining projects at the top level of the repository.

Sling Commons Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|----------|--------------|-------------------------------|---------------------------------------|
| OK | fmeschbe | commons/json | org.apache.sling.commons.json | JSON support from json.org |
| OK | fmeschbe | commons/mime | org.apache.sling.commons.mime | Generalized MIME type mapping service |

These projects in the `commons` folder contain common functionality used by multiple other projects and may be either referred to by way of OSGi package import or by inlining the respective packages/classes into the bundle.

JCR Related Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|----------|-----------------------|--|---|
| OK | fmeschbe | jcr/api | org.apache.sling.jcr.api | Bundle-ized JCR/Jackrabbit API + Session Pooling |
| OK | fmeschbe | jcr/resource | org.apache.sling.jcr.resource | JCR based ResourceResolver, Mapping (formerly the <code>sling-content-jcr</code> bundle in the <code>content</code> folder) |
| OK | fmeschbe | jcr/classloader | org.apache.sling.jcr.classloader | JCR based ClassLoader |
| OK | fmeschbe | jcr/jackrabbit-client | org.apache.sling.jcr.jackrabbit.client | Access existing JCR Repository |

| | | | | |
|----|----------|--------------------------|---|-----------------------------|
| OK | fmeschbe | jcr/jackrabbit-server | org.apache.sling.jcr.jackrabbit-server | Embedded JCR Repository |
| OK | fmeschbe | jcr/jackrabbit-extractor | org.apache.sling.jcr.jackrabbit-extractor | Jackrabbit Query Extensions |

The projects in the `jcr` folder are related to JCR.

Sling Launcher Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|----------|-----------------|----------------------------------|--|
| OK | fmeschbe | launcher/app | org.apache.sling.launcher.app | Command line Sling launcher (embedded Jetty) (formerly the <code>sling-app</code> project) |
| OK | fmeschbe | launcher/webapp | org.apache.sling.launcher.webapp | Sling launcher Web Application (formerly the <code>sling-servlet</code> project) |

Launchers for Sling are located in the `launcher` folder. Two projects currently exist. The `app` project which launches Sling from the command line and the `webapp` project which is a web application which launches Sling as a web application in a servlet container.

Maven Plugin Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|-----------|--------------------------|-------------------------------------|---|
| OK | ctiegeler | maven/maven-jcrom-plugin | org.apache.sling.maven-jcrom-plugin | JavaDoc Tag support for JCR OCM Mapping |
| OK | fmeschbe | maven/maven-jspc-plugin | org.apache.sling.maven-jspc-plugin | JSP Compiler for bundled JSPs |
| OK | ctiegeler | maven/maven-sling-plugin | org.apache.sling.maven-sling-plugin | Various helper goals |

The `maven` folder contains Maven 2 plugin projects which simplify development of Sling itself as well as applications extending Sling.

microsling Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|----------|-------------------|------------------------------------|--------------------------------|
| OK | fmeschbe | microsling | - | Container for microsling |
| - | - | microsling/core | org.apache.sling.microsling.core | microsling Core Implementation |
| - | - | microsling/webapp | org.apache.sling.microsling.webapp | microsling Web Application |

The `microsling` folder contains the `microsling` projects which is a very simple implementation of the Sling API. `microsling` is only statically extensible (as opposed to Sling, which is extensible dynamically at run time).

OSGi Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|----------|------------------|----------------------------------|--|
| OK | fmeschbe | osgi/log | org.apache.sling.osgi.log | LogService impl and Log API provider |
| OK | fmeschbe | osgi/assembly | org.apache.sling.osgi.assembly | Assembly packaging (this project will be removed in the future because it provides a subset of the functionality of the new Deployment Package Service contained in the OSGi Compendium Service Specification R 4.1) |
| OK | fmeschbe | osgi/obr | org.apache.sling.osgi.obr | OSGi Bundle Repository server |
| OK | fmeschbe | osgi/console-web | org.apache.sling.osgi.onsole.web | Web Based OSGi management console |

The projects in the `osgi` folder are used to extend the functionality of an OSGi framework. These projects are used to implement running Sling instances but do not directly influence Sling itself.

Scripting Engine Projects

| Status | Who | Folder | Artifact ID | Description |
|-------------|--------------------|----------------------|---------------------------------------|--|
| In Progress | fmeschbe | scripting/resolver | org.apache.sling.scripting.resolver | ScriptResolver implementation and Engine support |
| OK | fmeschbe | scripting/jsp | org.apache.sling.scripting.jsp | JSP (includes Jasper) |
| OK | ctiegeler/fmeschbe | scripting/jsp-taglib | org.apache.sling.scripting.jsp.taglib | JSP TagLib for Sling |

| | | | | |
|----|-----------|----------------------|---------------------------------------|---|
| OK | chiegeler | scripting/javascript | org.apache.sling.scripting.javascript | JavaScript using Rhino (includes ESP support) |
| OK | chiegeler | scripting/velocity | org.apache.sling.scripting.velocity | Velocity Templating |
| OK | chiegeler | scripting/freemarker | org.apache.sling.scripting.freemarker | Freemarker Templating |
| OK | chiegeler | scripting/ruby | org.apache.sling.scripting.ruby | Ruby Templating |
| - | - | scripting/??? | org.apache.sling.scripting.??? | more scripting engines, e.g. JSR-223 |

One of the most notable extensions of the Sling API compared to the Component API is the better definition of implementing scripting. Formerly scripting was supported as just a special way of a Component requiring more or less complex setup in the repository. The projects in the `scripting` folder implement the new `SlingScriptingEngine` interface for various script languages. The `resolver` project implements the `SlingScriptResolver` interface for Sling.

Sling Main Projects

| Status | Who | Folder | Artifact ID | Description |
|--------|-----------|-------------------------|------------------------------------|---|
| OK | fmeschbe | sling/core | org.apache.sling.core | The Core Sling engine. See note below. |
| - | - | sling/cache | org.apache.sling.cache | Output caching of response data (currently empty) |
| OK | chiegeler | sling/event | org.apache.sling.event | Distributed Eventing using JCR |
| OK | fmeschbe | sling/sample | org.apache.sling.sample | Sample Servlets and JSPs |
| OK | chiegeler | sling/scheduler | org.apache.sling.scheduler | Quartz-based Scheduler for Sling |
| OK | fmeschbe | sling/servlets-standard | org.apache.sling.servlets.standard | Standard Servlets, e.g. <code>nt:file</code> , <code>nt:folder</code> |

The projects in the `sling` folder are the core Sling API implementation projects. In addition added value projects such as scheduling and distributed eventing projects are also located in this folder.

Note on `sling/core`: The core project is currently almost a complete implementation of the Sling framework. Only the JCR Repository access and Object Content Mapping functionality is not part of the core. This project should be broken apart such that only the bare minimum of the functionality is contained. This is to be further analyzed.

Removed and Replaced Projects

- The `component-api` project defining the old Component API is replaced by the new `api` project defining the Sling API. Therefore the `component-api` project is removed.
- The project JSP compiler (Jasper plus Eclipse Java Compiler) project `jasper-sling` is not a standalone project any more. It has been integrated into the project implementing the `SlingScriptingEngine` for JSP `script/jsp`.