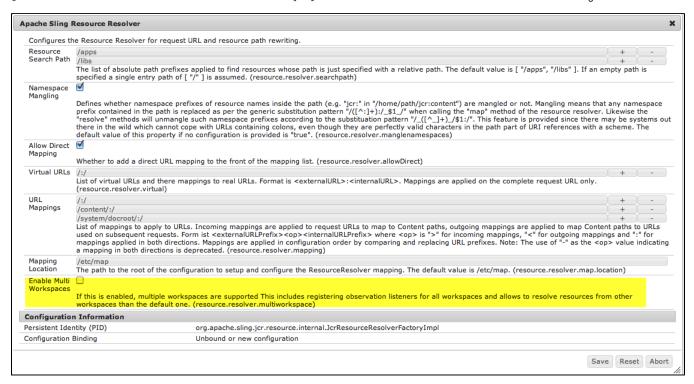# Multiple Workspace Support

Multi-Workspace Support

The Java Content Repository (JSR-170/JSR-283) standard requires that content repositories be composed of one or more workspaces which contain the actual content nodes. Workspaces are intended to be used to store different branches of the same node graph (see http://wiki.apache.org/jackrabbit /DavidsModel#Rule_.233:_Workspaces_are_for_clone.28.29.2C_merge.28.29_and_update.28.29.).

Resource Resolution

By default, only the default workspace will be consulted during resource resolution.

Configuration

To enable multi-workspace resource resolution, set the `resource.resolver.multiworkspace` configuration property of PID `org.apache.sling.jcr.resource.internal.JcrResourceResolverFactoryImpl` to true. This can be done via the Web Console on the Configuration screen.

By default, Sling's JCR Resource Resolver will resolve resources from the default workspace (typically named "default"). An alternate workspace can be specified by including a workspace as a colon-delimited prefix to the path. For example:

```
Resource res = resResolver.getResource("ws2:/path/to/node");
```

Request-based Resolving

When resolving a resource based on an `HttpServletRequest`, the workspace name can also be provided through a request attribute named `org.apache.sling.api.resource.ResourceResolver/use.workspace`. Note that Sling does not provide any mechanism to set this attribute. Since resource resolution occurs very early in the Sling request pipeline (see http://sling.apache.org/site/filters.html), you must set this attribute either outside of Sling (i.e. in a Servlet Filter) or in the Authentication phase.

If this attribute is set at the time of resolution (i.e. is passed to `ResourceResolver.resolve(HttpServletRequest)` or `ResourceResolver.resolve(HttpServletRequest, String)`), then future invocations of `ResourceResolver.adaptTo(Session.class)` will return a `javax.jcr.Session` logged into the requested workspace.

This request-bound workspace is also used for `findResources()` and `queryResources()`.
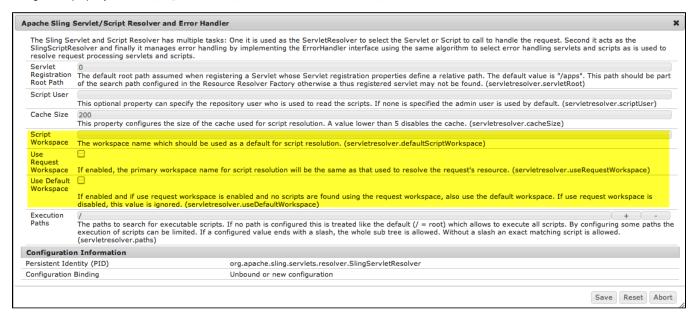
Impact on Resource instances

Regardless of how a Resource is resolved in a non-default workspace, the path of the resolved resource will contain the workspace name as a path prefix. This means that `Resource.getPath()` will not equal `Resource.adaptTo(Node.class).getPath()`.

JCR Resource Events

The jcr.resource bundle includes support for converting JCR Observation events into OSGi EventAdmin events. Once multi-workspace support is enabled, resource events will be sent for JCR events across all workspaces, with the resource path containing the workspace prefix.

Script Resolution

By default, scripts are always resolved from the default workspace, regardless of how the Resource Resolver is configured. The default workspace for scripts can be configured separately from the default workspace for resources by setting the `servletresolver.defaultScriptWorkspace` configuration property of PID `org.apache.sling.servlets.resolver.SlingServletResolver` to true.

Script resolution can also be configured to look first (or only) in the same workspace as the resource.

If `servletresolver.useRequestWorkspace` is true and `servletresolver.useDefaultWorkspace` is false, **only** the workspace containing the resource will be used for resolution.

If `servletresolver.useRequestWorkspace` is true and `servletresolver.useDefaultWorkspace` is also true, the resource's workspace will be consulted first and, if nothing is found, the default workspace will be checked.