OSGi 4.1 Overview
Short Talk

BJ Hargrave
OSGi Alliance CTO
IBM Lotus
Version 4.1 - An incremental improvement

- A number of minor enhancements, clarifications and errata fixes.
- The 2 major changes in Version 4.1 occurred in the life cycle layer.
  - Adding a transient flag to the Bundle start and stop methods
  - Adding a lazy activation mode to Bundle start.
Persistent Start and Stop

- Normally when `Bundle.start()` or `Bundle.stop()` are called, the framework must persistently remember the autostart setting of the bundle.

- That is, if a bundle is started with `Bundle.start()`, then, when the framework is next restarted, that bundle will automatically be restarted.*

* Assuming the bundle's startlevel is met
Transient Start and Stop

- The Bundle start and stop methods have now been overloaded with new variants that take option flags
  - See the new `Bundle.start(int)` and `Bundle.stop(int)` methods

- Two of these option flags control whether the start or stop action is transient
  - That is, whether the framework must alter the persistent autostart setting of the bundle

- If the transient option is specified, then the framework does not alter the persistent autostart setting when performing the start or stop operation

New in 4.1
“Eager” Activation

- Normally, when Bundle.start is called, the bundle will be activated*
  - That is, the BundleActivator is called
- During framework launch starting many bundles can result in a large number of bundles being activated slowing down the launch process
- Many of these bundles don’t actually need to be activated during launch
- They just need to be activated prior to being used

* Assuming the bundle's startlevel is met
Lazy Activation

- Allows the activation of a bundle to be delayed until the first class load is made from the bundle.
- The caller of Bundle.start can ignore the lazy activation policy declared in the bundle's manifest by using a new start option.
  - This allows an admin agent to require "eager" activation even when the manifest says "lazy".
- Finally, there is a new bundle event type that can be received by SynchronousBundleListeners to advise them that a bundle has started and is lazily awaiting activation.
  - This can be used by extender bundles like OSGi Declarative Services, Spring OSGi and iPOJO to receive notification that the bundle has been started and now has a valid BundleContext.
  - The extender bundle can then act on behalf of the bundle and do things like register services for the bundle.

New in 4.1
Download Version 4.1 Now

- The spec is now final and published
- It can be downloaded from
  http://www2.osgi.org/Download/Release4V41