Item 221
git_comments:

git_commits:

1. **summary:** ZOOKEEPER-1334. Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed (Claus Ibsen via phunt)
   message: ZOOKEEPER-1334. Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed (Claus Ibsen via phunt) git-svn-id: https://svn.apache.org/repos/asf/zookeeper/trunk@1423778 13f79535-47bb-0310-9956-ffa450edef68

github_issues:

github_issues_comments:

github_pulls:

github_pulls_comments:

github_pulls_reviews:

jira_issues:

1. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
   label: test
2. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
3. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version>

<scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

4. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
   label: code-design
5. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
6. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import

packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

7. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

8. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

   label: code-design

9. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

   description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

10. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId>

<version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

11. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

12. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

13. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

14. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

15. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
16. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
17. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="

[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

18. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
19. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
20. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
21. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId>

<version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

22. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
23. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
24. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
25. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed

description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

26. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
27. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
28. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="

[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

29. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
30. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
31. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion=" [1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.
32. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId>

<version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

33. **summary:** Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed
    description: In Zookeeper 3.3.x you use log4j for logging, and the maven dep is eg from 3.3.4 {code} <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} Now in 3.4.0 or better you changed to use slf4j also/instead. The maven pom.xml now includes: {code} <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-api</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>org.slf4j</groupId> <artifactId>slf4j-log4j12</artifactId> <version>1.6.1</version> <scope>compile</scope> </dependency> <dependency> <groupId>log4j</groupId> <artifactId>log4j</artifactId> <version>1.2.15</version> <scope>compile</scope> </dependency> {code} But the META-INF/MANIFEST.MF file in the distribution did not change to reflect this. The 3.3.4 MANIFEST.MF, import packages {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} And the 3.4.1 MANIFEST.MF, import packages: {code} Import-Package: javax.management,org.apache.log4j,org.osgi.framework;v ersion="[1.4,2.0)",org.osgi.util.tracker;version="[1.1,2.0)" {code} This makes using zookeeper 3.4.x in OSGi environments not possible, as we get NoClassDefFoundException for slf4j classes.

jira_issues_comments:

1. **body:** thanx for reporting this. what would be a good way of testing this bug so that it doesn't happen in the future?
   label: test
2. The OSGI manifest is hardcoded in your build.xml https://svn.apache.org/repos/asf/zookeeper/trunk/build.xml You would need to add {code} org.slf4j;version="[1.6,2.0)" {code} To include the slf4j dependency
3. Benjamin, Testing OSGi is not super easy, but there is a OSS project called Pax-Exam that has an OSGi based test kit. We use this for other Apache project such as: Camel, ActiveMQ, etc. There is a pax exam 2.x release which is even better. We have not yet upgrade from 1.x to 2.x in Camel, as its a bit work to do. The 2.x should be much faster and allows to do in-jvm debugging etc. The 1.x is slow and it fork a jvm for testing. And it downloads a lot of JARs etc. http://team.ops4j.org/wiki/display/paxexam/Pax+Exam That was the reason we discovered this as we upgraded camel-zookeeper from 3.3.4 to 3.4.1 and our osgi based test failed. The regular junit test passed of course. At Apache Camel we have 2 kind of osgi tests - karaf feature test [1] - osgi unit tests [2] The former is just a test that the camel components can be installed in an osgi runtime such as Apache Karaf. This test will frankly bootup Apache Karaf and then do an osgi install of the bundles. That way we can often spot 3rd party JARs which is not OSGi compliant. The latter is real full blown osgi unit tests, where we use the components and do something. For a quick test, you may in Zookeeper, have the first kind of test, where you will install zookeeper as a bundle in an osgi runtime, such as apache karaf (it can test with using felix, equinox as osgi framework). 1) - https://svn.apache.org/repos/asf/camel/trunk/tests/camel-itest-karaf/ 2) - https://svn.apache.org/repos/asf/camel/trunk/tests/camel-itest-osgi/
4. **body:** And if you no longer uses log4j directly in zookeeper, then i suggest to mark its maven dependency as <scope>test</scope>. Then end users of zookeeper who uses maven do not depend on log4j. They may use another logger such as logback or java util logging etc. Also if so then the import of org.apache.log4j should be removed from the MANIFEST.MF for OSGi end users.
   label: code-design
5. Hi guys, I'm gonna work on this issue to "fix" ZooKeeper in an OSGi way.
6. Jean-Baptiste I'll assign to you then. Thanks. Also take a look at ZOOKEEPER-1078 where we're working on Maven support, I suspect that will impact any changes you make to the current ant based codebase?

7. Another issue I am seeing with the Manifest is that package org.apache.zookeeper.server.persistence is not exported. {code:xml} Export-Package: org.apache.zookeeper;version="3.4.3",org.apache.zookee per.data;version="3.4.3",org.apache.zookeeper.version;version="3.4.3" ,org.apache.zookeeper.server;version="3.4.3",org.apache.zookeeper.ser ver.quorum;version="3.4.3" {code}
8. **body:** Ah - my bad - I see that package org.apache.zookeeper.server.persistence should not be exported. One of our libs is trying to import this package unnecessarily. Please disregard previous comment.
   label: code-design
9. Any update on this?
10. Any update. Dont the Apache Zookeeper project want to be OSGi compliant? Seems like a fairly easy fix in your ANT build script.
11. I gonna review and submit a patch if required. Thanks for the reminder Claus.
12. Just testing ZooKeeper 3.5.0 and it works in Karaf. In Karaf, I have: Bundle-Vendor = The Apache Software Foundation Bundle-Name = ZooKeeper Bundle Bundle-DocURL = http://hadoop.apache.org/zookeeper Bundle-SymbolicName = org.apache.hadoop.zookeeper Bundle-Version = 3.5.0 Bundle-License = http://www.apache.org/licenses/LICENSE-2.0.txt Bundle-ManifestVersion = 2 Import-Package = javax.management, org.apache.log4j, org.osgi.framework;version="[1.4,2.0)", org.osgi.util.tracker;version="[1.1,2.0)" Export-Package = org.apache.zookeeper;version=3.5.0, org.apache.zookeeper.data;version=3.5.0, org.apache.zookeeper.version;version=3.5.0, org.apache.zookeeper.server;version=3.5.0, org.apache.zookeeper.server.quorum;version=3.5.0
13. However: - slf4j import should be present (and optional) - log4j import should be optional
14. If you completely switched the code to use the slf4j-api. Then you should not have compile time deps to slf4j-log4j12 and log4j. These should be scope test. For compile scope you should only depend on slf4j-api. In the Manifest you should also only import the slf4j api packages not log4j. Then it will work great in apache karaf.
15. I found another problem with the Manifest. org.apache.zookeeper.server.ZooKeeperServer depends on org.apache.zookeeper.server.persistence.FileTxnSnapLog but the package is not exported. So I propose to also export the package: org.apache.zookeeper.server.persistence
16. Any update of this. Christian or Jean it seems the Zookeeper team is not keen on helping with this. Maybe if we provide a .patch file that they can just apply. It prevent us from using Zookepper 3.4.x or newer releases in OSGi. In fact if the Zookeeper team is *NOT* listening to the community, it would be better to remove all the OSGi bits from their MANIFEST.MF file. Then at least the release is NOT attempting to be OSGi bundles. But of course its a fairly easy fix, and we can help test it works fine in OSGi, ... just that the Zookeeper team need to listen to the community.
17. To be clear, we are keen on seeing this fixed, however we don't have sufficient experience with osgi to drive this. If you want to see it fixed, something that is known to work in your environment, providing a patch that we can review and commit is the best way to get this addressed.
18. Is this patch ready for commit review, or still work needs to be done? (recent comments seem to indicate still some issues?
19. A patch to the build.xml which fixes the osgi manifest issues.
20. zookeeper-1334-osgi.patch is the file with the patch.
21. Installing zookeeper 3.5.0 JAR in Karaf 2.3.0 works fine now with that patch {code} [58] [Active] [ ] [80] ZooKeeper Bundle (3.5.0) [59] [Active] [ ] [80] The Netty Project (3.5.1.Final) karaf@root> headers 58 ZooKeeper Bundle (58) -------------------- Manifest-Version = 1.0 Main-Class = org.apache.zookeeper.server.quorum.QuorumPeer Ant-Version = Apache Ant 1.8.2 Built-By = davsclaus Built-At = 12/19/2012 05:44 GMT Built-On = davsclaus.lan Implementation-Version = 3.5.0--1 Implementation-Vendor = The Apache Software Foundation Implementation-Title = org.apache.zookeeper Created-By = 1.6.0_37-b06-434-11M3909 (Apple Inc.) Bundle-Vendor = The Apache Software Foundation Bundle-Name = ZooKeeper Bundle Bundle-DocURL = http://hadoop.apache.org/zookeeper Bundle-SymbolicName = org.apache.hadoop.zookeeper Bundle-Version = 3.5.0 Bundle-License = http://www.apache.org/licenses/LICENSE-2.0.txt Bundle-ManifestVersion = 2 Import-Package = javax.management;resolution:=optional, org.slf4j;version="[1.6,2)", org.jboss.netty.buffer;resolution:=optional;version="[3.2,4)", org.jboss.netty.channel;resolution:=optional;version="[3.2,4)", org.jboss.netty.channel.group;resolution:=optional;version="[3.2,4)", org.jboss.netty.channel.socket.nio;resolution:=optional;version="[3.2,4)", org.osgi.framework;resolution:=optional;version="[1.5,2)", org.osgi.util.tracker;resolution:=optional;version="[1.4,2)" Export-Package =

org.apache.zookeeper;version=3.5.0, org.apache.zookeeper.client;version=3.5.0, org.apache.zookeeper.data;version=3.5.0, org.apache.zookeeper.version;version=3.5.0, org.apache.zookeeper.server;version=3.5.0, org.apache.zookeeper.server.auth;version=3.5.0, org.apache.zookeeper.server.persistence;version=3.5.0, org.apache.zookeeper.server.quorum;version=3.5.0 {code}

22. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12561647/zookeeper-1334-osgi.patch against trunk revision 1422772. +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. -1 patch. The patch command could not apply the patch. Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1308//console This message is automatically generated.
23. Updated patch - use --no-prefix with git to create patch.
24. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12561655/zookeeper-1334-osgi.patch against trunk revision 1422772. +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1309//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1309//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1309//console This message is automatically generated.
25. Committed to 3.4.6 and trunk. Thanks Claus!
26. -1 overall. Here are the results of testing the latest attachment http://issues.apache.org/jira/secure/attachment/12561655/zookeeper-1334-osgi.patch against trunk revision 1422772. +1 @author. The patch does not contain any @author tags. -1 tests included. The patch doesn't appear to include any new or modified tests. Please justify why no new tests are needed for this patch. Also please list what manual steps were performed to verify this patch. +1 javadoc. The javadoc tool did not generate any warning messages. +1 javac. The applied patch does not increase the total number of javac compiler warnings. +1 findbugs. The patch does not introduce any new Findbugs (version 1.3.9) warnings. +1 release audit. The applied patch does not increase the total number of release audit warnings. +1 core tests. The patch passed core unit tests. +1 contrib tests. The patch passed contrib unit tests. Test results: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1310//testReport/ Findbugs warnings: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1310//artifact/trunk/build/test/findbugs/newPatchFindbugsWarnings.html Console output: https://builds.apache.org/job/PreCommit-ZOOKEEPER-Build/1310//console This message is automatically generated.
27. Integrated in ZooKeeper-trunk #1777 (See [https://builds.apache.org/job/ZooKeeper-trunk/1777/]) ZOOKEEPER-1334. Zookeeper 3.4.x is not OSGi compliant - MANIFEST.MF is flawed (Claus Ibsen via phunt) (Revision 1423778) Result = SUCCESS phunt : http://svn.apache.org/viewcvs.cgi/?root=Apache-SVN&view=rev&rev=1423778 Files : * /zookeeper/trunk/CHANGES.txt * /zookeeper/trunk/build.xml
28. I've tested this patch on a locally build zookeeper (3.4.5 + patch from ZOOKEEPER-1334) using Karaf 2.2.8 and 2.3.0 (both Equinox and Felix) with a simple test bundle that creates a ZooKeeper connection upon the bundle being started. In my tests this patch only works for the combination Karaf 2.2.8 + Equinox. In the other scenario's I ran into some NoClassDefFoundErrors on classes located in the packages {{javax.security.auth.callback}} , {{javax.security.auth.login}} and {{javax.security.sasl}} . I resolved this by adding these packages to the Import-Package section of the ZooKeeper bundle. @Claus: should these packages be added to the Import-Package section of the MANIFEST of the ZooKeeper bundle or are users expected to expose these packages to ZooKeeper bundle through OSGi framework configuration?
29. Hi [~arnoud]. Please create a new jira for this issue, the original one has already been resolved. Thanks!
30. [~arnoud.buitenhuis@sogeti.nl] No the MANIFEST.MF should contain all the packages that this bundle will or may use. So the 3 packages you found should be added as well.

31. Okay, I've created a new issue for this: ZOOKEEPER-1645
32. Closing issues after releasing 3.4.6.