Fuse IDE
Release Notes

Version 2.0
July 2011

Updated: 15 Nov 2011

New Features

The following are new Fuse IDE features in 2.0:

OSGi Blueprint Support
The route editor can now read and write OSGi Blueprint files.

JMX Explorer
Fuse IDE discovers all of the processes deployed as part of your integration applications using JMX. You can inspect all of the properties published by the services.

Endpoint Browsing
Inspect the contents of Apache Camel endpoints that support browsing.

JMS Browsing
Inspect the contents of JMS destinations without consuming messages.

Route Tracing
Fuse IDE allows you to debug routes by tracing messages as they pass through a deployed route.

Enhancements

The following enhancements were made to Fuse IDE in 2.0:

Route editor drag and drop performance
The route editor has been enhanced to improve the drag and drop performance.

Route editor properties
The route editor's property editor has been enhanced to make setting properties easier.

Route validation
The route editor's validation capabilities have been improved to make it easier to round trip routes. It will now provide warnings when attempting to read or write invalid XML DSL.

Added Maven Archetypes

The **New Fuse Project** wizard includes entries for all known archetypes Fuse ESB, Fuse Message Broker, Apache Camel, and Apache CXF.

Known issues

The following are issues are know to exist in Fuse IDE 2.0:

Fuse IDE does not work in Spring STS

Fuse IDE cannot be installed into Spring STS eclipse distribution. There is a version clash between the version of m2eclipse used by Spring and the version required by Fuse IDE.

Upgrades do not always succeed

On occasion updating Fuse IDE will fail. To recover you will need to uninstall Fuse IDE, as described in "Removing Fuse IDE" in *Installation Guide* , and then reinstall Fuse IDE, as described in "Installing Fuse IDE" in *Installation Guide*.

No JDK warning

By default Eclipse uses your system's `PATH` setting to determine which Java VM to use. If you have multiple Java VM's installed on your system, this can result in Eclipse choosing an unsupported VM.

We recommend explicitly configuring your Eclipse installation to use the VM from a Java 6 JDK. You configure the VM used by Eclipse by adding the following two lines to your `eclipse.ini` file:

```
-vm
JDKInstallDir/bin/javaw.exe
```

J2SE-1.5 execution environment

New projects may generate with a `Build path specifies execution environment J2SE-1.5. There are no JREs installed in the workspace that are strictly compatible with this environment.` warning. To resolve the warning, you need to remove the J2SE-1.5 JRE library from the build path and replace it with a Java 6 JRE library.