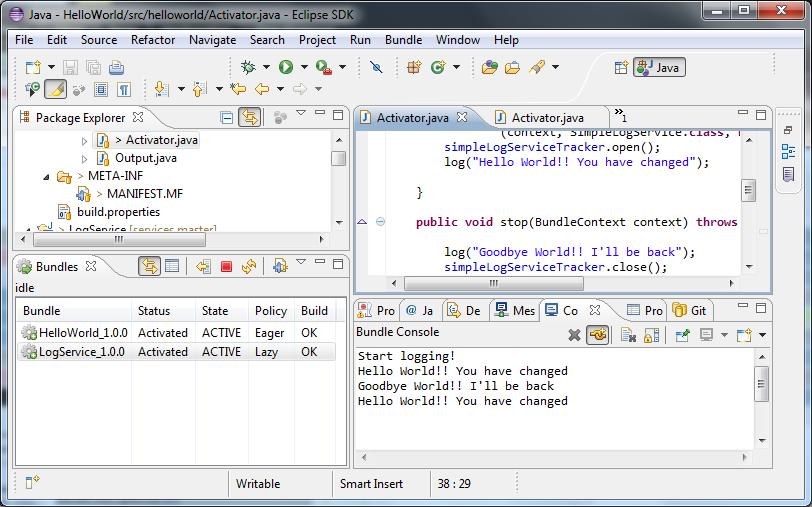
Workspace Bundles Overview

# Purpose

The main job of the InPlace Activator is to activate and update the run time instances of the source plug-in projects you decide to activate. Bundles are installed, resolved and started when a project is activated and updated each time an activated project is built. An activated bundle responds to changes in the plug-in project and to OSGi commands. All OSGi commands are available, either explicit or implicit, and it is possible to set options for activation policy and define dependency rules for bundles being activated, deactivated, started and stopped

# The Bundles View

In the following screenshot two bundles are activated; - A log service with a lazy activation policy; and - a hello world bundle logging messages through the log service at start and stop. Activation of the hello world bundle force the lazy activated log service bundle to start as can be seen from the ACTIVE state in the *Bundles View*.



# Install the InPlace Activator and Activate Bundles

In addition to get status of workspace bundles and issue bundle commands for individual bundles in the *Bundles View* a global Bundle menu is provided to execute bundle commands that operates on all or a subset of the plug-ins in the workspace. Bundle commands, including the activate command, bundle options and dependency settings are available from the main menu. Error messages and warnings are sent to the *Bundle Console View*.

The rest of this overview shortly describes how the most used bundle commands works. To actually try out the different commands you can walk through one of the tutorials in the Getting Started section in the help system or just create a new bundle from the *New Plug-in Project* wizard and start out by activating the generated plug-in project.

# Activate Bundles

When activating a bundle it is first installed, resolved and lastly started, where the start method of the activator of the bundle is invoked. If the bundle requires capabilities from other deactivated bundles, they are activated implicit. After a bundle is activated it is possible to explicit start, stop, refresh and reset the bundle in addition to deactivate it.

# Build and update Bundles

An activated bundle is updated when its plug-in project is built. An update implies stopping and unresolving the bundle before the compiled source of the plug-in project of the bundle is updated, refreshed (or optionally resolved) and started again.

# Deactivate Bundles

When you deactivate a bundle it either becomes uninstalled or if there are other active bundles in the workspace it is left in state installed. Deactivated bundles can only be activated and does not respond to OSGi commands from within the IDE or when its corresponding source project is built.

# Bundles with Dependencies

When a bundle is activated all bundles that provide capabilities to a bundle being activated is activated as well. When a bundle is deactivated, all bundles that directly and indirectly require capabilities from the bundle to deactivate are also deactivated. Beside this you can use dependency options to tailor whom dependent bundles are going to be activated/deactivated and started/stopped.

# Bundles with Build Errors

When the source project of an activated bundle has built errors the bundle is not updated and the current revision of the bundle is used. It is not possible to activate a project with build errors.

# A Deactivated Workspace

When all bundles are deactivated, the workspace is said to be deactivated. In this state all bundles are in state uninstalled and the InPlace Activator does not respond to any changes in the workspace besides updating the state (if e.g. an external OSGi command is issued) and status fields in the *Bundles View*.