Project and Bundle Operations

Contents

[Project and Bundle Relationships 1](#_Toc335574308)

[An active Workspace 1](#_Toc335574309)

[Explicit and Implicit Bundle Operations 1](#_Toc335574310)

[Project and Workspace Commands 1](#_Toc335574311)

[The Relationship between Project and Bundle Operations 2](#_Toc335574312)

This section presents an overview of the collection of project and workspace wide IDE commands and their relationship to bundle operations.

A bundle operation or command affects the life cycle of a bundle by moving the bundle from one state to another. See [Bundle States and Transitions](../concepts/Dynamic%20Workspace%20Bundles.htm#BundleStatesAndTransitions) for an overview of the possible transitions and states a bundle may take.

## Project and Bundle Relationships

There is a conditional *one-to-one* (1:0-1) part of relationship between a project and a workspace bundle. The relationship is uniquely defined by the project name and one of the bundle identifiers, which for a workspace bundle, is the combination of the symbolic name and the version as defined in the manifest file. The part-of relationship is physically implemented by the fact that the manifest file is part of the project identified by the project name. The project and the bundle also share a common location identifier, which is used to install the bundle from the project. When a bundle is in state UNINSTALLED the relationship is (1:0), and (1:1) for all other bundle states.

## An active Workspace

Due to this strong relationship between a project and a bundle, project wide operations, as e.g. renaming and moving a project, affects activated bundles in different ways.

In a deactivated workspace all bundles are in state UNINSTALLED. A deactivated workspace is passive in the sense that it is not reacting to changes in the workspace - project operations and builds - and there are no bundle commands but the activate command available.

## Explicit and Implicit Bundle Operations

You change the state of a bundle by executing bundle commands (e.g. start and stop) or by certain project/workspace operations (e.g. save or close a project).

The standard OSGi operations are, install, uninstall, update, resolve, refresh, start and stop. For workspace bundles, activate and deactivate are added as composite bundle commands performing multiple OSGi operations on a bundle.

All commands are further divided into explicit (activate, deactivate, reset, refresh, start, stop) and implicit (install, uninstall, update, resolve) commands. Explicit commands are executed directly from the UI. Implicit and explicit commands are executed by the InPlace Activator when project and workspace commands are executed.

## Project and Workspace Commands

Project and workspace wide commands that triggers bundle operations are:

* **Project:** Create, import, open, delete, close, rename, move and project build (manual and automatic) commands.
* **Workspace:** All manual and automatic commands that affect more than one project. Workspace build or a Save all are examples of workspace wide commands.

A build command triggers a bundle update operation. All project commands are considered CRUD operations. The operations are Create (create new, open and import), Rename (rename and move), Update (save when automatic build is on, manual build and updated manually) and Delete (delete and close) a project.

Activated bundles respond automatically to CRUD operations on projects. Create, Rename and Update are recognized as changes and an automatic or manual build is needed to get a new compiled project state for bundle operations to react on.

* **Create** (create new, open and import) projectinstalls a bundle, moving it to state INSTALLED, if the workspace is activated, otherwise the bundle stays in state UNINSTALLED. If a project is opened or imported, the bundle is started if the project was activated when exported or closed.
* **Rename** (rename and move) project reinstalls an activated bundle, by first uninstalling the bundle with the original name (rename) or location (move) and then installing, resolving and starting the bundle with the new name or location.
* **Update** (save when automatic build is on, manual build and updated manually) of a modified project that has been built stops, updates, resolves and starts the corresponding activated bundle and dependent bundles again in the right dependency order.
* **Delete** (delete and close) a project deactivates a bundle by uninstalling it before the project is deleted and removed from the workspace. If a project is closed it is uninstalled before it is hidden from the workspace. Note that if a deleted or closed project is referenced from other requiring projects, the next build fails.

The goal of an activated workspace is to keep the activated bundles up to date where the runtime instances reflects the most recent builds of projects. If possible bundles are restored to the same state as they had before any CRUD operation.

## The Relationship between Project and Bundle Operations

To illustrate the interaction between IDE projects operations and OSGi bundle commands the implicit update command may serve as an example.

If *Build* *Automatically* is switched on, *File* | *Save* (project update operation), after a modification of some content in a project, triggers a build which again triggers a bundle update command. If the bundle was active prior to the save command, the bundle is first stopped and then unresolved. The updated version of the bundle is then read from the input stream and installed by replacing the old content. Lastly the bundle is resolved and started again. Update creates a new bundle revision and if the [Refresh on Update](Setting%20Bundle%20Options.htm#RefreshOnUpdate) option is on any unused revisions and their wires are released.

If the bundle to update has changed its requirements to other bundles by importing packages from say a deactivated bundle, the deactivated (or providing) bundle is activated and started as part of the update process.