Bundle Project Errors

Contents

[Bundle Projects 1](#_Toc347957436)

[Bundle Errors and Project Build Errors 1](#_Toc347957437)

[Errors in Deactivated, Activated and Disabled Projects 1](#_Toc347957438)

[Project Enabling and Bundle Relationship Errors 1](#_Toc347957439)

[Error Conditions Summary 2](#_Toc347957440)

# Bundle Projects

A bundle project refers to both the project and the bundle as one coherent unit. Build errors are related to the project (e.g. source code and project description errors) and are therefore also bundle errors and vice versa.

# Bundle Errors and Project Build Errors

Because a build error also is a bundle error it prohibits a bundle to resolve. Special for bundles are duplicate bundles; - that is when two bundles have the same symbolic name and version. For duplicates only one of the bundles can be installed. A bundle duplicate error is not defined as a project build error by the build system.

# Errors in Deactivated, Activated and Disabled Projects

Bundle errors can occur while a bundle is deactivated or after the bundle is activated. A bundle must be error free before it can be activated. Errors in activated bundles are not updated and the current revision and wires is used until the bundle becomes error free and either updated implicit or manually.

Disabled projects are source projects that have not yet been read and loaded by the Eclipse IDE. Projects become disabled at shut down and when they are deleted, closed and exported (the exported copy). A project becomes enabled at start up and when it is created (or re-created), opened or imported. Errors in a disabled activated project is treated as a separate error category when the project is enabled as explained in the [Project Enabling and Bundle Relationship Errors](#_Project_Enabling_and) section.

# Project Enabling and Bundle Relationship Errors

Of special interest are build errors with circular and broken references between bundle projects. A bundle project has by definition errors if the bundle project itself has errors or any of the bundle projects it references (or requires capabilities from) has errors, does not exist or is disabled. If the project indirectly references itself it’s a circular reference error.

It is not possible to activate a bundle, or any bundle requiring capabilities from the bundle, with errors. When errors occur in an already activated bundle, the bundle and the requiring bundles are not updated and the current revisions with the existing wires are used and the states of the bundles are not affected. When errors are corrected and the bundle project is built, the bundle and its requiring bundles are updated either manually or automatically.

If a bundle project has errors when it is exported, closed or when eclipse is shut down it is not possible to resolve the bundle at startup or when the bundle is imported or opened. Instead, the activated bundle with errors is moved to state INSTALLED if it is a build error and stays in state UNINSTALLED if the bundle is a duplicate of another activated bundle.

Deleting and closing bundle projects causes requiring projects (broken reference) to be unresolved (moved to state INSTALLED) in an activated workspace and uninstalled in a deactivated workspace.

# Error Conditions Summary

In summary, deviation from the general rule depicting that only deactivated bundles are moved to state UNINSTALLED or state INSTALLED happens under the following error conditions for activated bundles:

* At session start up, when the workspace is activated and when projects are opened or imported.
  + If an activated bundle has errors (except duplicate errors) it is moved to state INSTALLED. The bundle is resolved and possible started when the errors are corrected.
  + If the activated bundle has the same symbolic name and version (duplicate bundle) as an already activated bundle it is left in state UNINSTALLED. The bundle is installed, resolved and possibly started when the bundle has a unique combination of symbolic name and version among the activated bundles (workspace and deployed bundles).
* When a providing bundle project is closed or deleted (broken reference).  
  When closing or deleting a bundle that provides capabilities to one or more activated bundles the requiring bundles are moved to state INSTALLED. The requiring bundles are resolved and possible started again when the providing bundle project is opened, recreated, or the requirement (e.g. removing the providing bundle from the import header) to the providing bundle is removed.

# Runtime Errors and Program Logic

The Start and Stop commands executes code by invoking the start and stop method of their bundle. This may introduce different kind of runtime errors. One such fatal error is when a bundle runs out of memory. Control is not returned from the bundle responsible for the error and the status handler in Eclipse report the error in a modal dialog box. If the bundle continues allocating memory, a new dialog box reporting the same error is shown almost instantly after quitting the current error reporting dialog box. To avoid the deadlock the InPlace Activator overrides the standard status handler an only sends the error message to the log view, giving the user the option to terminate the IDE from the Bundle Console toolbar. Next time the IDE is started activated bundles are resolved but not started. This gives the user a chance to correct the error before starting the bundle.

A value of “Bundle Problems” in the *Status* field in the Bundle View indicates a bundle runtime error.