

Removing and disabling plugins

Jenkins allows you to **disable** a plugin as well as **uninstall** a plugin. What are the difference?

Uninstalling a plugin

Uninstalling a plugin removes the plugin binary (*.jpi or *.hpi) from the disk. The plugin continues to function normally until you restart Jenkins, but once you restart, Jenkins will behave as if you didn't have the plugin to being with. They will not appear anywhere in the UI, all the extensions they contributed will disappear.

Uninstalling a plugin, however, does not remove the configuration that the plugin might have created. If there are existing jobs/slaves/views/builds/etc that used some extensions from the plugin, during the boot Jenkins will report that there are some fragments in those configurations that it didn't understand, and pretend as if it didn't see such a fragment.

Because the configuration is still intact until you overwrite those configuration files, at this point if you re-install the uninstalled plugin and restart Jenkins, the configuration will come back.

If you want to scrab the configuration left by uninstalled plugins, go to "Manage Jenkins" and click "Manage Old Data".

Disabling a plugin

Disabling a plugin is a softer way to retire a plugin. Jenkins will continue to recognize that the plugin is installed, but it will not start the plugin, and no extensions contributed from this plugin will be visible.

The fragments contributed from a disabled plugin to configuration files would follow the same fate as in the case of uninstalled plugins.

Plugins can be disabled from the UI (or by touch `$JENKINS_HOME/plugins/$PLUGIN_NAME.jpi.disabled` (or `$JENKINS_HOME/plugins/$PLUGIN_NAME.hpi.disabled`) file.)

Plugins that are disabled can be re-enabled from the UI (or by removing `*.jpi.disabled` (or `*.hpi.disabled`) file from the disk.)