

MavenToolchains

This is a reference for the Maven Toolchains descriptor.

The default location for the toolchains file is `~/.m2/toolchains.xml`

A Toolchain is a preconfigured object that Maven plugins can use for tool configuration retrieval (location and other information).

The [toolchains-plugin](#) can read available toolchains on the user's computer and match them against the toolchain requirements of the project (as configured in `pom.xml`): if match is found, the toolchain instance is made available to other Maven plugins.

With `jdk` toolchain, for example, instead of being stuck with the JDK used to run Maven, all plugins can use the same other JDK instance without hardcoding absolute paths into the `pom.xml` and without configuring every plugin that require path to JDK tools.

See the [Guide to Using Toolchains](#) for more information.

```
<toolchains xmlns="http://maven.apache.org/TOOLCHAINS/1.1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/TOOLCHAINS/1.1.0 http://maven.apache.org/xsd/toolchains-1.1.0.xsd">
  <toolchain>
    <type/>
    <provides>
      <key>value</key>
    </provides>
    <configuration/>
  </toolchain>
</toolchains>
```

toolchains

The `<toolchains>` element is the root of the descriptor. The following table lists all of the possible child elements.

Element	Type	Description
toolchain *	List<ToolchainModel>	(Many) The toolchain instance definition.

toolchain

Definition of a toolchain instance.

Element	Type	Description
type	String	Type of toolchain: <ul style="list-style-type: none">• <code>jdk</code> for JDK Standard Toolchain,• other value for Custom Toolchain (Many) Toolchain identification information, which will be matched against project requirements. For Maven 2.0.9 to 3.2.3, the actual content structure was completely open: each toolchain type would define its own format and semantics. In general, this was a properties format. Since Maven 3.2.4, the type for this field has been changed to Properties to match the de-facto format. Each toolchain defines its own properties names and semantics. Toolchain configuration information, like location or any information that is to be retrieved.
provides/key=value*	Properties	Actual content structure is completely open: each toolchain type will define its own format and semantics. In general, this is a properties format: <name>value</name> with per-toolchain defined properties names.
configuration	DOM	