

# Configuring External Dependencies

This topic explains how to configure external dependencies when custom extensions depend on Java libraries other than `com.parasoft.api.jar`.

Custom extensions can depend on Java libraries other than `com.parasoft.api.jar`. For example, assume that you want your extension to use an open source Java library. In order for SOAtest or Virtualize to successfully load such an extension, it must also locate the dependencies. To enable loading of that extension, you must list any jars used by your extension (excluding `com.parasoft.api.jar`) in a "ClassPath" header in your extension jar's MANIFEST.MF file. The Java tutorial provides details about how to set up a jar to depend on other jars (see <http://docs.oracle.com/javase/tutorial/deployment/jar/downman.html>).

Alternatively, you may package your extension and its dependencies into a single combined jar file. This is useful if you want to distribute your extension as a single jar instead of a collection of jars. If you are building your custom extension with Apache Maven, then you can automate this task with the `maven-shade-plugin` (see <http://maven.apache.org/plugins/maven-shade-plugin/>).

Custom extensions should *not* depend on internal, non-public jars. This includes any jar that is installed by SOAtest or Virtualize, but is not in `com.parasoft.api.jar`. Any extensions that depend on such jars are unsupported and may not function in a later release.