

# Settings

---

There are two places where Osmorc's settings can be changed:

- Project specific and application wide settings are managed on the [OSGi](#) page of the [Settings](#) dialog box.
- Module specific settings are managed through the [Osmorc module facet settings](#).

## Project and application settings

The [OSGi](#) page

## Facet settings

In the facet settings you first have to decide whether you will edit the manifest files yourself or whether you want Osmorc to create it with bnd from the dependencies on other modules and libraries.

When you decide to maintain the manifest file manually, Osmorc creates dependencies on modules and the framework instance bundles in the module where the manifest file is changed. Here you take charge of the OSGi dependencies between your bundles and Osmorc tries to match them with corresponding module dependencies.

Letting Osmorc generate the manifest files for you is exactly the other way around. You develop your application as you would develop any other application in IntelliJ IDEA. You add dependencies on other modules and libraries and Osmorc tries to translate those dependencies into OSGi dependencies when running your bundles.

If you are maintaining the manifest file yourself, you have to tell Osmorc where it is to be found. You can change this location for each module or you can define a project default in the project settings and use it here.

Each time a module is compiled, Osmorc creates a bundle *JAR* for that module. This *JAR* is used when running an Osmorc run configuration. You can change how the *JAR* is named and where it is put, but for most cases the created default should be OK.

The created *JAR* contains the compilation output and the manifest file - either the manually maintained or the one created by Osmorc. Sometimes you will need to add some other files that need to be copied into the generated *JAR*. For Eclipse RCP bundles - or plugins as they are called there - you will need to add the `plugin.xml`. You can do this in the table listing additional *JAR* content. The first column takes the path to the source file relative to the content root of the module and the second column takes the path to the destination file. So since the `plugin.xml` of an Eclipse RCP plugin normally resides in the root of a plugin, you will type in `plugin.xml` for both columns of that entry. Future versions will provide a file chooser to choose the source file.

Currently the table for the additional *JAR* contents is very simple. It grows automatically as you type in new file definitions. There is always an empty row at the bottom that waits for new content. If you want to remove an entry, simply remove both the source and destination definition for it. It is then deleted automatically.

## See Also

Reference:

- [OSGi Facet Page](#)
- [OSGi](#)

Web Resources:

- [Developer Community](#) 

