

# Converting XNAT 1.6 Modules to XNAT 1.7+ Plugins



The XNAT team has developed a tool to help with migrating XNAT 1.6 modules into 1.7 plugins. [Source Code / Documentation: https://bitbucket.org/xnatdev/module\\_to\\_plugin](https://bitbucket.org/xnatdev/module_to_plugin)

*This is an excerpt from the module conversion documentation on Bitbucket.*

## What the script does

### Re-organize your files

The first thing the script does is re-organize the structure of the files and folders in your module. The structure of files in 1.7 plugins is different than in 1.6 modules, so things must be organized a little differently. Here is a list of the directories you might have in your module and where those files will end up in your plugin:

- `src/java` -> `src/main/java`
- `src/schemas` -> `src/main/resources/schemas`
- `src/templates` -> `src/main/resources/META-INF/resources/templates`
- `src/xnat-templates` -> `src/main/resources/META-INF/resources/xnat-templates`
- `src/xdat-templates` -> `src/main/resources/META-INF/resources/xdat-templates`
- `src/scripts` -> `src/main/resources/META-INF/resources/scripts`
- `src/style` -> `src/main/resources/META-INF/resources/style`
- `src/images` -> `src/main/resources/META-INF/resources/images`
- `repository` -> `src/main/repository`

In addition to looking for files in directories under `src`, the script will also look in directories under `src/main` and `src/main/resources/module-resources`.

### Inspect your code for likely problems

The next thing the script does is look through the Java code itself, looking for bits of XNAT code that you may be using and that we know have changed. Some XNAT classes and methods changed from 1.6 to 1.7. If the script detects that some of your Java classes are trying to use code that will likely no longer work in 1.7, the script will warn you and provide instructions for how you can change your code so it will work in 1.7. The script will not edit your source code during this stage, just tell you where the problems are likely to be.

We have tried to anticipate XNAT changes that may affect your modules and add warnings for them, but we can't possibly cover every change that was made to XNAT code that you may be using. The warnings produced by the script are not comprehensive; this means that just because there aren't any warnings is no guarantee that all of your Java code will work with 1.7. If you encounter any issues getting Java classes working in 1.7, please [contact us](#). We will help you figure out why your code is not working and may add more warnings to the conversion script to help other people avoid the issue you ran into.

### Create a gradle build script

By default, XNAT 1.7 plugins are built using the [Gradle build tool](#). The script creates a `gradlew` gradle wrapper script and a `build.gradle` file pre-loaded with the most essential dependencies. If you run the script with the `-b` flag set, then a longer list of common dependencies will be included. Your plugin will likely only need a subset of the commonly used dependencies, so you may wish to use a tool like [Gradle Lint](#) to remove dependencies you do not use. If your module is built with an existing build tool like maven or gradle, your build files will not be copied over, so you should update the `build.gradle` file manually to make sure that any of these dependencies that you still need are included.

If you are new to using gradle as a build tool, you may find their documentation helpful. For instance, see their guide on [using an existing gradle build](#) and [migrating from maven](#).