

# Dependency Injection

When Ashcroft fails a test, you will see that the recommended solution often involves a refactoring towards dependency injection. Fine, that (with the aid of [Mock Objects](#)) will allow you to test your class in isolation. But what about integration tests? Or even the whole application? How do you assemble the **real** classes (including these low level ones) that actually **do** access files, sockets, threads and graphics environments?

There are several ways. You can do it manually in some bootstrap class, or you can use PicoContainer/NanoContainer or some other DI framework such as Spring or HiveMind. It's up to you. These frameworks will make that assembly easy, as long as your classes honour DI.