Legacy Fedora Module Lifecycle

There is work underway to shift Fedora modules to work as Spring beans. This information will be out of date when that occurs, but here are some basics about it works today (as of Fedora 3.4-RC1):

Most of Fedora's pluggable functionality has historically been written in terms of Java interfaces, which can be swapped out and configured via fedora.fcfg <module> elements.

Module Lifecycle

When Fedora starts:

1. fedora.fcfg is read
2. An instance of the appropriate fcfg-configured subclass of fcrepo.server.Server is created. No one has ever written an alternative subclass of Server, so this is actually always BasicServer.
3. BasicServer.initServer is called, which:
    a. Creates a single instance of each <module..> class found in the configuration.
    b. Then, in no particular order it calls each Module's initModule method
    c. Finally, again, in no particular order, it calls each Module's postInitModule method.

The proper way for a module to get a reference to another module is to call: getServer().getModule(String role). If such a module exists, this returns a Module instance that can then be cast to the appropriate functional interface (the "role", as it's called in fedora.fcfg)

When Fedora stops:

1. BasicServer calls each module's shutdownModule method. If an exception is thrown, it is logged, and the remaining module's shutdownModule methods are called.
2. BasicServer.shutdown is called.