



Seam Configuration and Administration

The title of this chapter isn't very compelling, but that's for a reason: this chapter discusses the dull but critical details related to the configuration of Seam applications. Before you can jump in and start writing applications with Seam, you need to set up your application deployment package with the necessary Seam libraries and configure your application code to "plug in" the Seam capabilities.

Much as you might want to ignore these issues, they are essential to actually working with Seam. A book with the title *Practical JBoss™ Seam Projects* would be remiss if it didn't cover these topics, and you'd be negligent if you didn't get familiar with them to some degree. Luckily, I've figured out a lot of the hiccups in installing, configuring, and debugging Seam applications for you, so read on to avoid learning these lessons all over again.

Preparing the Application Server

Seam expects certain standard APIs and services to be available in the application server where a Seam application is going to run. Depending on the type and version of application server you plan to use, the task of preparing the application server to run a Seam application can be more or less involved.

In this section, I'll first describe some basic requirements of Seam, and then discuss some specific environments and issues specific to getting Seam applications to run in these environments.

Java 5.0 Required

Regardless of the particular server you plan to use, Seam requires a Java 5.0 runtime, at a minimum. You can't use any of the annotations provided by JBoss Seam, or by EJB 3.0, in a Java 1.4 environment. If your application server does not operate in a Java 5.0 environment or isn't "certified" in that environment, you might want to reconsider using Seam, or switch to a different application server to run your Seam applications.

Before you do any of the installation I describe in the upcoming text, make sure you have a current Java 5.0 runtime installed on your system, and ensure that your application server is configured to use the Java 5.0 runtime. Java 5.0 for Linux, Windows, and Solaris can be found on the Sun Java site (<http://java.sun.com>). Apple provides their own Java runtime implementations for Mac OS X, and they are available through the Software Update tool on OS X.

JavaServer Faces

Seam integrates many of its core services (contextual components, web controls, etc.) into the JSF runtime environment. At the time of this writing, Seam is compatible with both JSF 1.1 and 1.2. A compliant Java EE 5.0 environment requires a JSF 1.2 implementation, but several “pre-Java EE” environments, such as JBoss 4, have either JSF 1.1 or 1.2 implementations bundled into their configurations. The JBoss 4.0.x releases, for example, use Apache MyFaces as their JSF implementation, and MyFaces is only compliant with JSF 1.1. But Sun has released a full reference implementation of JSF 1.2 that can be integrated into your application server if needed, and the Seam facilities are compatible with it.

Enterprise JavaBeans 3.0

Technically, it is possible to run JBoss Seam without an EJB 3.0 container. You’ll be eliminating some of the interesting features of Seam, such as the use of session beans as JSF managed beans, but you can still make use of Seam’s contextual component services, and many of the other Seam integrated features, such as jBPM and pageflow.

If your application server includes an EJB 3.0 container, you shouldn’t have to do anything special to configure it for use with Seam. If your server doesn’t have an EJB 3.0 container, you have a few choices, as explained next.

Running Seam with the JBoss Embedded EJB 3.0 Container

You can use the JBoss embedded EJB 3.0 container with Seam in your application. The embedded EJB 3.0 container can be embedded directly into your application (hence the name), allowing you to run EJB 3.0 components regardless of the application server capabilities. The advantage of this is that you don’t need to alter any application server configuration details, which could be an issue in some situations. You may be running multiple applications on your server, for example, some of which need an EJB 2.1 environment. You could also be running in an environment where an operations group tightly controls the application server configuration (for very good reasons), and significant changes like this are difficult to implement globally.