JBoss™ The Professional Open Source Company

The JBoss 4 Installation Guide

JBoss AS 4.0.4

Release 1

Table of Contents

About this Guide

JBoss AS is a J2EE 1.4-certified, open source Java application server. It is the most widely used application server on the market. The highly flexible and easy-to-use server architecture has made JBoss AS the ideal choice for users just starting out with J2EE, as well as senior architects looking for a customizable middleware platform. The ready availability of the source code allows you to debug the server, learn its inner workings and create customized versions for your personal or business use. This guide will show you how to install JBoss AS 4.0. You will learn how to start and stop your JBoss instance, and you will also learn about the directory structure and understand what the key services and configuration files are.

1

Installing JBoss

Before installing the server, you need to check your system to make sure you have a suitable Java installation. JBoss 4 requires either a Java 1.4 or Java 5 (sometimes referred to as Java 1.5) JVM (Java Virtual Machine) to run. Java 5 is only required to use the newer simplified EJB3 technologies. The choice of JVMs is yours otherwise, but we do recommend considering the Java 5 JVM where possible to take advantage of the latest JVM performance improvements and monitoring capabilities. No matter what JVM is chosen, you should generally prefer the latest stable versions and keep an eye on future bug fix releases for issues that might affect your installation.

To verify your Java environment execute the `java -version` command. This will ensure that the `java` executable is in your path and that you are using the intended Java version. The following output shows a Java 5 (sometimes called Java 1.5) JVM. If you don't see the appropriate version, check your JVM installation instructions.

```
$ java -version
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_02-56)
Java HotSpot(TM) Client VM (build 1.5.0_02-36, mixed mode, sharing)
```

JBoss AS is distributed as part of the Red Hat JBoss Media Kit. Recent JBoss releases are also available (both in binary and source form) online from the JBoss AS downloads page, http://labs.jboss.org/portal/jbossas/download/, and as part of the JEMS distribution, available online at http://labs.jboss.com/portal/jemsinstaller/downloads.

JBoss is released in two forms. The first is a ZIP or tgz archive containing a base JBoss installation. We'll talk about the ZIP version, as any system containing a Java installation will have the tools to extract the archive. The gzipped tar file requires a gnutar-compatible tar program that can handle the long pathnames in the archive. The default tar binaries on Solaris and OS X do not currently support the long pathnames.

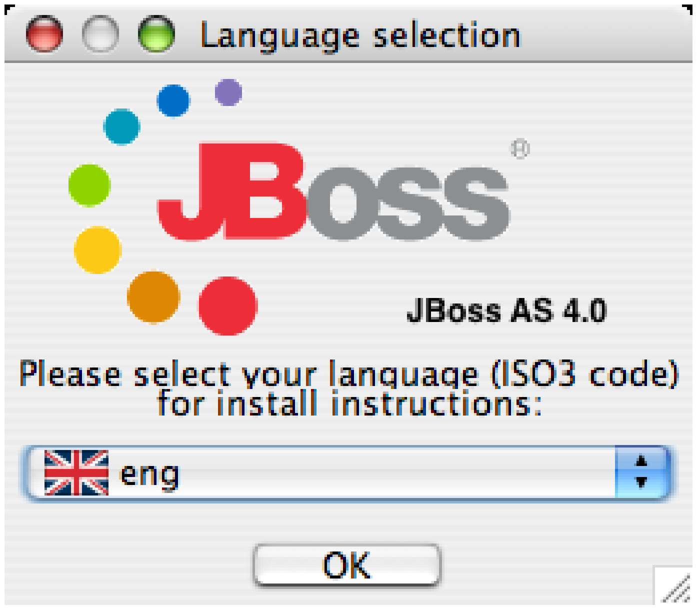
The standard JBoss 4.0.4 ZIP distribution is named `jboss-4.0.4.GA.zip`. If service pack releases later become available, they will contain the `SP` designation, `jboss-4.0.4SP1.zip` for example. You can use the JDK `jar` tool (or any other ZIP extraction tool) to extract the archive contents into a directory of your choice. It does not matter where on your system you install JBoss. Note, however, that installing JBoss into a directory that has a name that contains spaces causes problems on some platforms with Sun-based VMs.

The resulting JBoss installation is a raw installation containing the entire set of JBoss AS services in a completely unconfigured state. It is the quickest way to get a runnable JBoss instance, but the latter work to configure the server can be quite intensive. JBoss now provides a GUI installer that can simplify the installation process. In addition to the basic installation, the installer allows you to select the which services are installed, ensuring all service dependencies are met, secure the installation and configure a default datasource. Using a custom JBoss install created by the installer can greatly simplify the installation and configuration of JBoss.

The installer can be run directly from a web browser using Java Web Start or can be downloaded as an executable JAR file. If you choose the Java Web Start option, you only need to click the Run Installer link for the desired JBoss version on the downloads page. The Java Web Start installer is quick and easy. However, options like command-line install will require manually downloading and running the installer executable JAR. On many operating system, you can run executable JARs by double-clicking them. If your system doesn't support that, you can run the installer directly from the command line: (The remainder of this guide will assume the JEMS 1.2.0 installer, available from http://labs.jboss.com/portal/jemsinstaller/downloads)

```
$ java -jar jems-installer-1.2.0.jar
```

When you launch the installer, you will be given the option to select the installer language as shown in Figure 1.1. This screen only selects the language that the installer will display choices in and has no effect on the language used by JBoss or the applications deployed in JBoss.

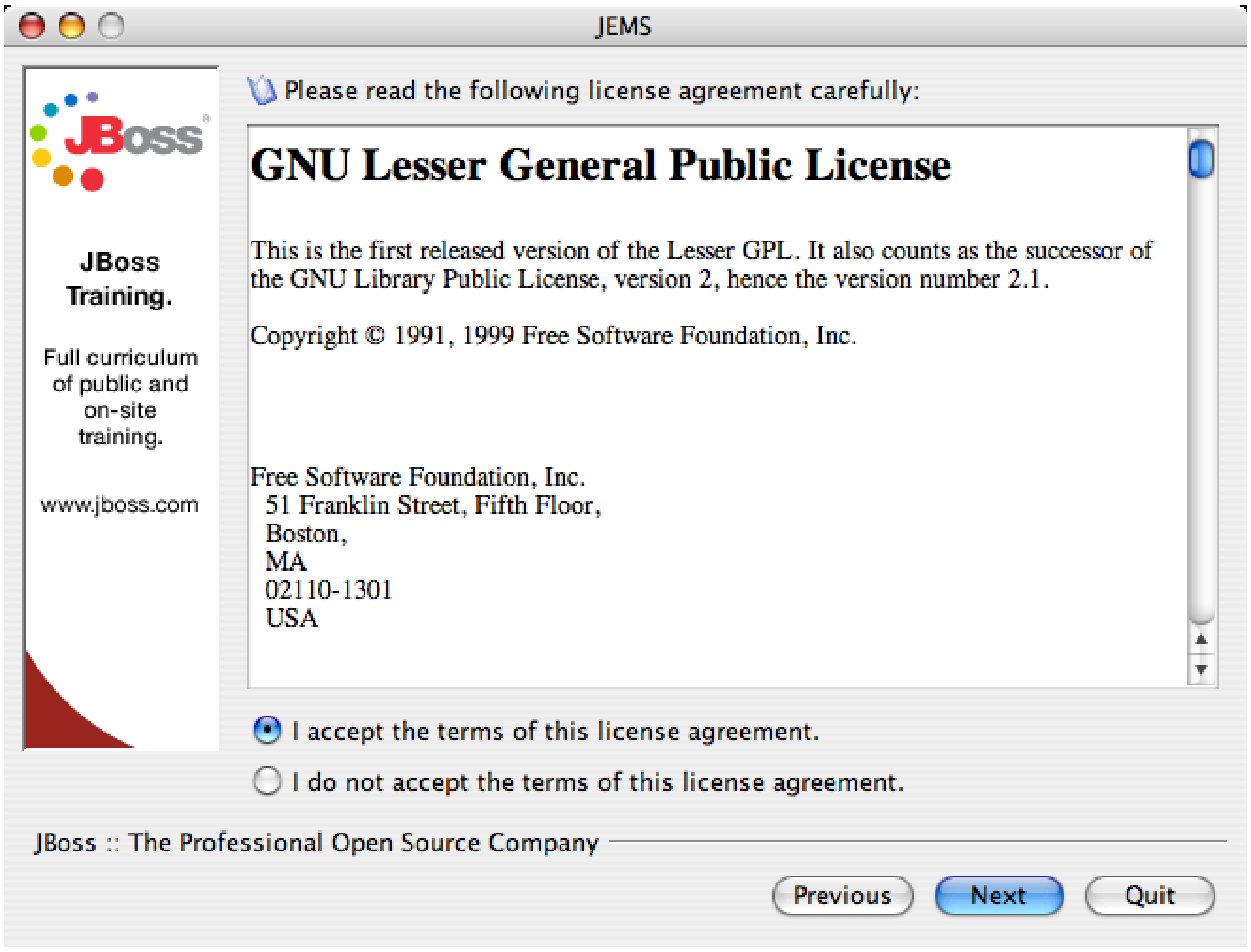


Please select your language (ISO3 code) for install instructions:

eng

Figure 1.1. The installer language selection screen.

After language selection, the installer will present a series of screens presenting the release notes and asking you to accept the JBoss license. JBoss is completely open source and is released under the GNU LGPL license. The license screen (shown in Figure 1.2) displays the full text of the LGPL license. More information on why JBoss uses the LGPL license and the advantages the LGPL provides to JBoss users can be found at http://www.jboss.com/company/licensing.



Please read the following license agreement carefully:

GNU Lesser General Public License

This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.

Copyright © 1991, 1999 Free Software Foundation, Inc.

Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor,
Boston,
MA
02110-1301
USA

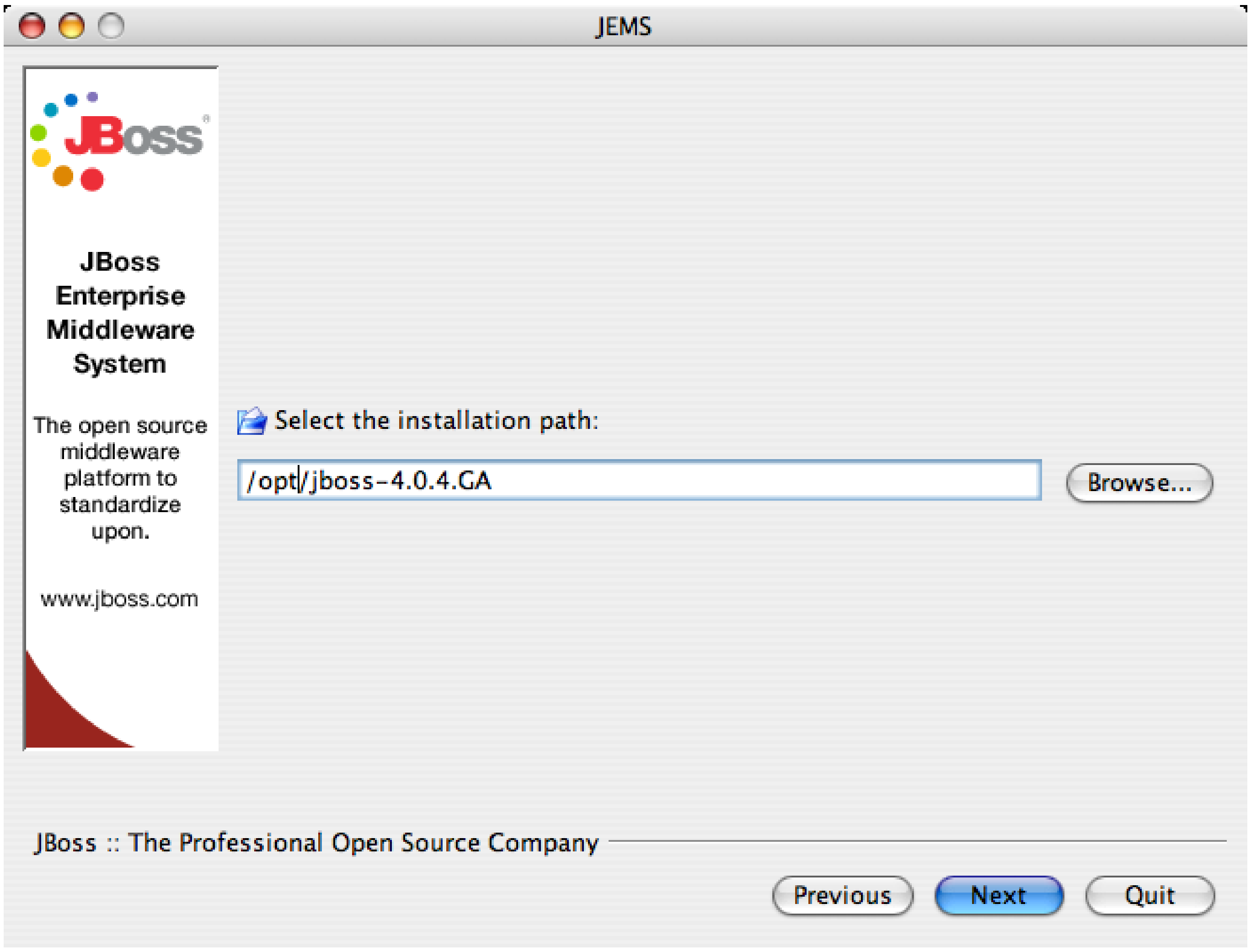
[✓ I accept the terms of this license agreement.]

I do not accept the terms of this license agreement.

JBoss :: The Professional Open Source Company

Figure 1.2. The license screen

The installer will ask for the directory to use for the installation. This is shown in Figure 1.3. The installer does not write the installation directory into any of the scripts or into any form of registry, so you will be free to move or rename the JBoss installation directory after installation. On some platforms, installation directories that contains spaces can cause problems, so we recommend sticking to simple directory names.

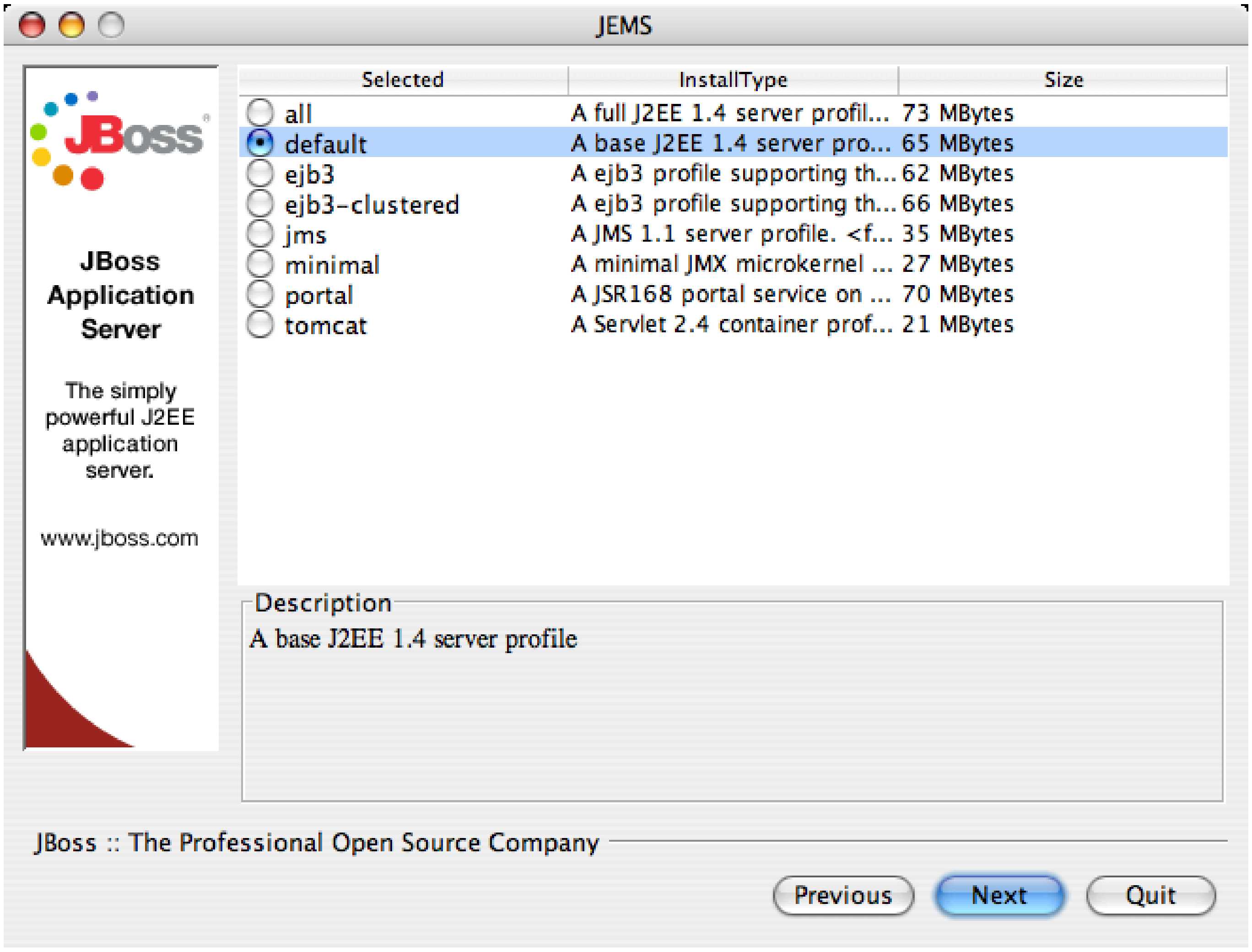

JEMS
JBoss
JBoss Enterprise Middleware System
The open source middleware platform to standardize upon.
www.jboss.com
Browse...
JBoss :: The Professional Open Source Company
Previous
Next
Quit

Select the installation path:

/opt/jboss-4.0.4.GA

Figure 1.3. Selecting the installation directory

After that you able to select the starting server configuration set, as shown in Figure 1.4.



Selected	InstallType	Size
all	A full J2EE 1.4 server profil...	73 MBytes
[✓ default]	A base J2EE 1.4 server pro...	65 MBytes
ejb3	A ejb3 profile supporting th...	62 MBytes
ejb3-clustered	A ejb3 profile supporting th...	66 MBytes
jms	A JMS 1.1 server profile. <f...	35 MBytes
minimal	A minimal JMX microkernel ...	27 MBytes
portal	A JSR168 portal service on ...	70 MBytes
tomcat	A Servlet 2.4 container prof...	21 MBytes

Description
A base J2EE 1.4 server profile

JBoss :: The Professional Open Source Company

Figure 1.4. Selecting the installation group

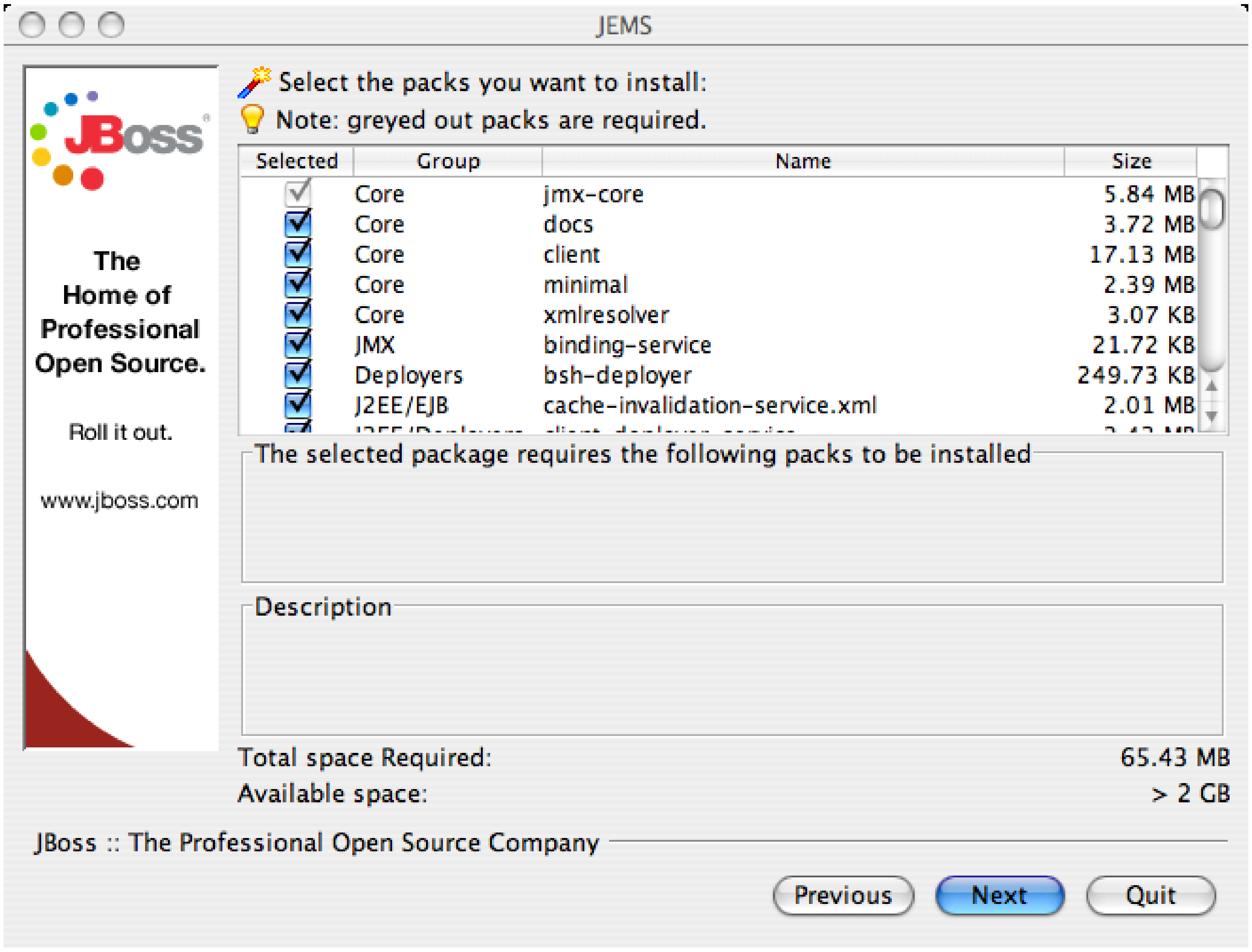
The starting configuration determine which sets of packages are available for installation. The following table describes each of the configuration sets.

Table 1.1. The JBoss AS installer configuration sets

Name	Description	Notes
`all`	A full J2EE 1.4 server profile with enterprise extensions such as clustering and IIOP.	
`default`	A base J2EE 1.4 server profile.	
`ejb3`	An EJB3 profile supporting the full EJB3 specification with Tomcat	This requires a Java 5 runtime and is not a J2EE 1.4 compatible configuration.
`ejb3-clustered`	An EJB3 profile supporting the full EJB3 specification with Tomcat and clustering.	This requires a Java 5 runtime and is not a J2EE 1.4 compatible configuration.

Name	Description	Notes
`jms`	A JMS 1.1 server profile	This is not a J2EE 1.4 compatible configuration.
`minimal`	A minimal JMX microkernel	This is not a J2EE 1.4 compatible configuration.
`portal`	A JBoss Portal 2.4 profile	This is not a J2EE 1.4 compatible configuration.
`tomcat`	A Servlet 2.4 container profile	This is not a J2EE 1.4 compatible configuration.

After selecting the configuration set, you have the option to further customize the installation, customizing the set of services installed. Figure 1.5 shows the package selection screen. The installer knows the dependencies between services and will not allow you to configure services in an incompatible way. This is much safer than the trial and error approach of configuring services by hand from a raw ZIP install. When choosing configuration sets, be aware that you can not add packages that are not a part of the selected configuration set. If you wanted a simple web container (the tomcat configuration) that also had JMS support (the jms configuration), it would be necessary to go to a larger configuration, such as the default configuration, and remove the unwanted packages. There are some combinations of JEMS components that are not supported directly through the installer.



Select the packs you want to install:
Note: greyed out packs are required.

Selected	Group	Name	Size
✓	Core	jmx-core	5.84 MB
✓	Core	docs	3.72 MB
✓	Core	client	17.13 MB
✓	Core	minimal	2.39 MB
✓	Core	xmlresolver	3.07 KB
✓	JMX	binding-service	21.72 KB
✓	Deployers	bsh-deployer	249.73 KB
✓	J2EE/EJB	cache-invalidation-service.xml	2.01 MB

The selected package requires the following packs to be installed

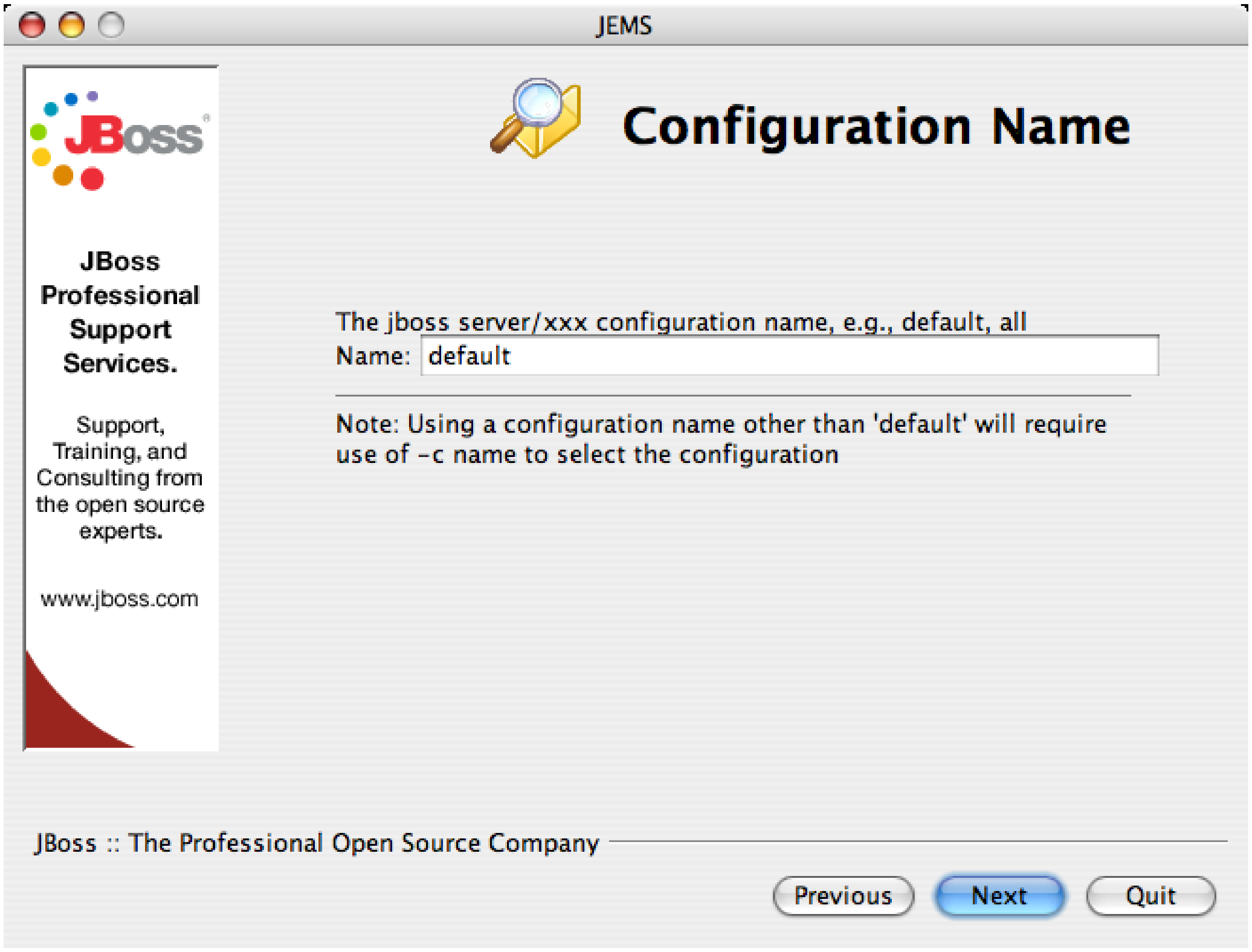
Description

Total space Required: 65.43 MB
Available space: > 2 GB

JBoss :: The Professional Open Source Company

Figure 1.5. Selecting the packages to install

The following screen (Figure 1.6) allows for the customization of the server configuration name. Unless you need to create multiple configurations, you should use a configuration name of `default`. Using any other configuration name requires you to start JBoss with the `-c` option to specify the configuration JBoss should use.


JEMS
JBoss
JBoss Professional Support Services.
Support, Training, and Consulting from the open source experts.
www.jboss.com
JBoss :: The Professional Open Source Company
Previous
Next
Quit

Configuration Name

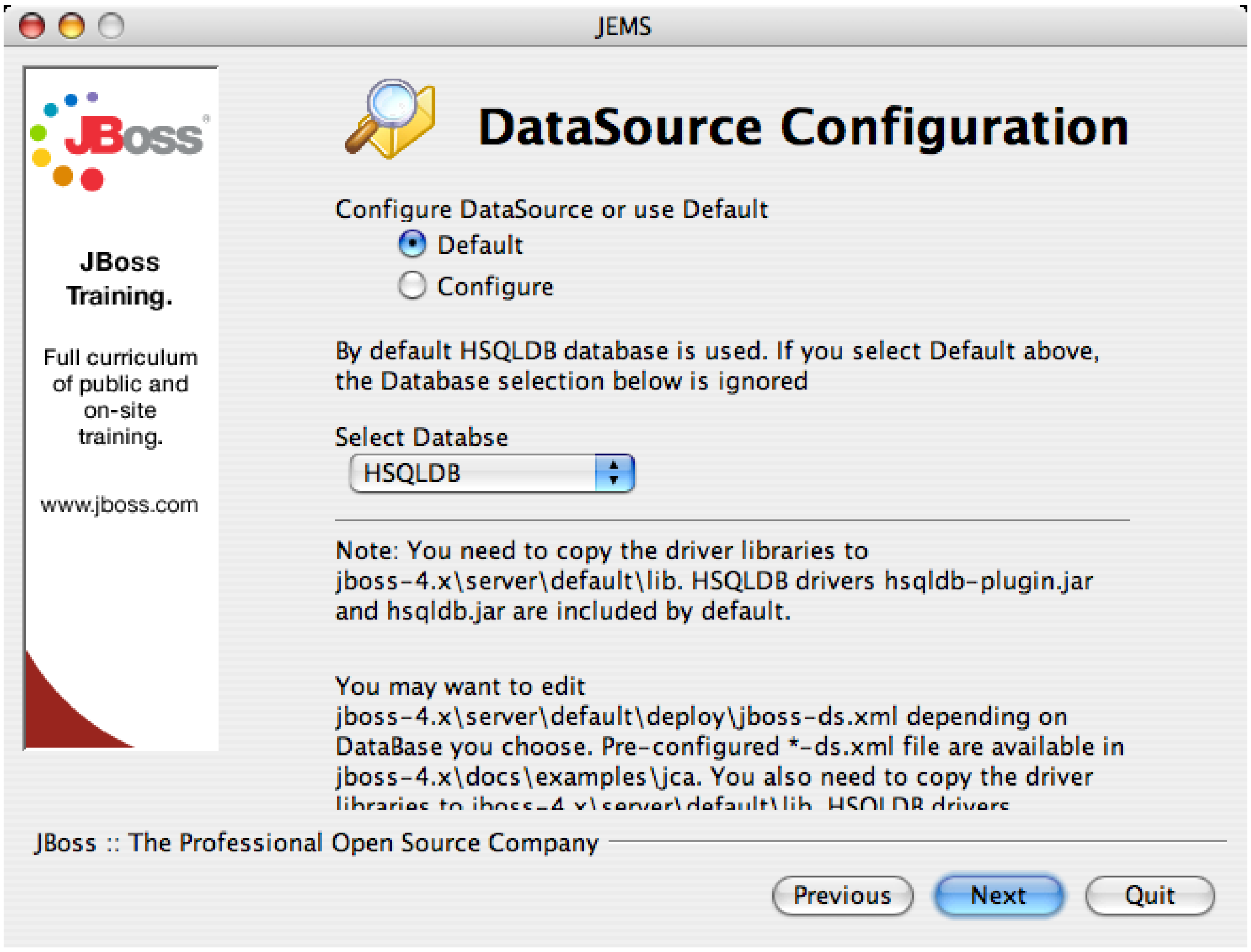
The jboss server/xxx configuration name, e.g., default, all

Name: default

Note: Using a configuration name other than 'default' will require use of -c name to select the configuration

Figure 1.6. Name the configuration

Almost all applications require a datasource to connect to a back-end database. JBoss provides an embedded Hypersonic database along with a default datasource to connect applications to. Being able to run applications out of the box makes JBoss very developer friendly. All projects will eventually need to move to a more capable database, but most will choose to do so at the very beginning of the project. The datasource configuration screen, shown in Figure 1.7, gives you the option to use the default hypersonic datasource to configure a replacement datasource.



DataSource Configuration

Configure DataSource or use Default

[✓ Default]

Configure

By default HSQLDB database is used. If you select Default above, the Database selection below is ignored

Select Databse

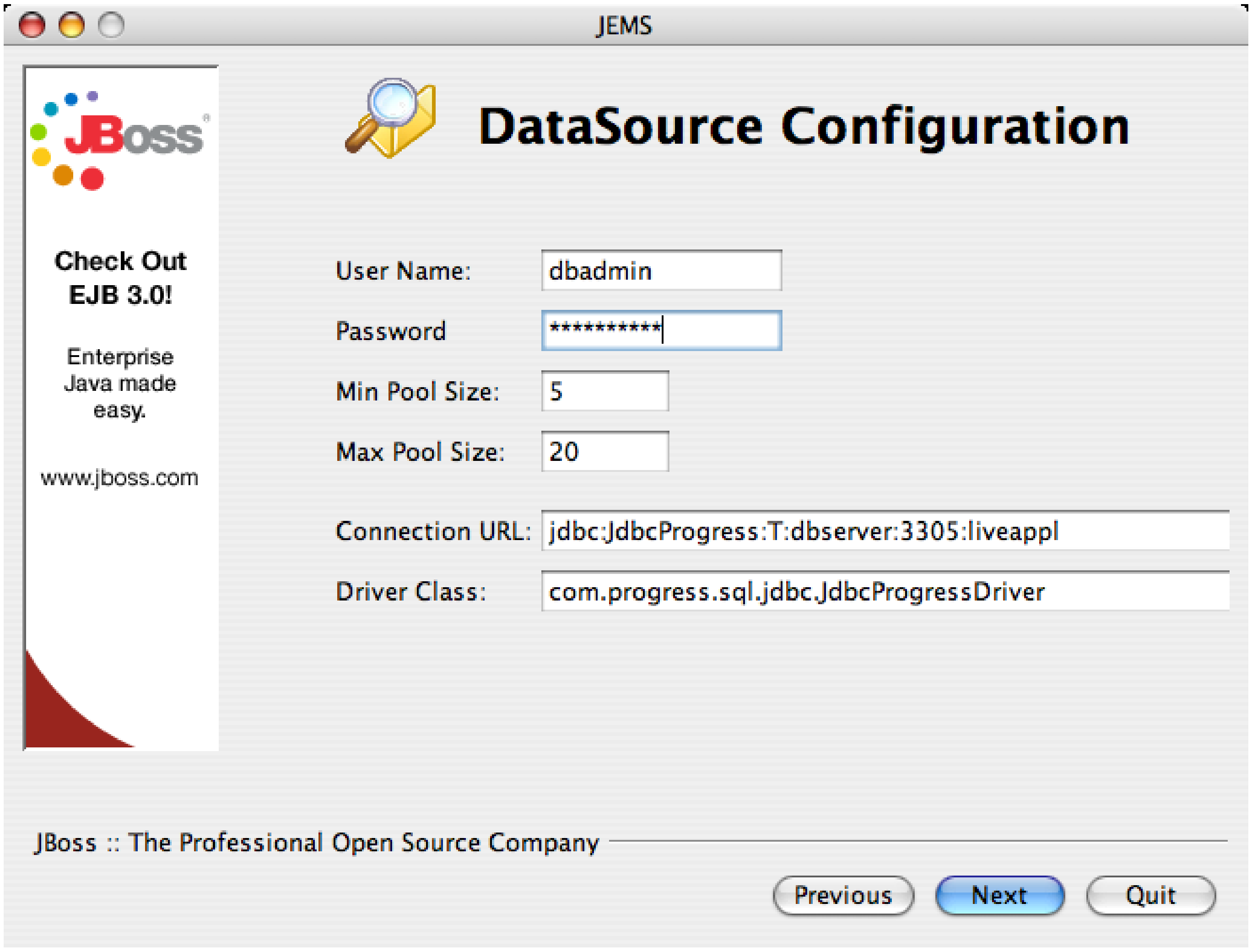
HSQLDB

Note: You need to copy the driver libraries to jboss-4.x\server\default\lib. HSQLDB drivers hsqldb-plugin.jar and hsqldb.jar are included by default.

You may want to edit jboss-4.x\server\default\deploy\jboss-ds.xml depending on DataBase you choose. Pre-configured *-ds.xml file are available in jboss-4.x\docs\examples\jca. You also need to copy the driver libraries to jboss-4.x\server\default\lib. HSQLDB drivers

Figure 1.7. Configure the default datasource

If you want to configure the datasource, choose the configure option and select the database type. You'll then be given the opportunity to enter database connection information as shown in Figure 1.8.



DataSource Configuration

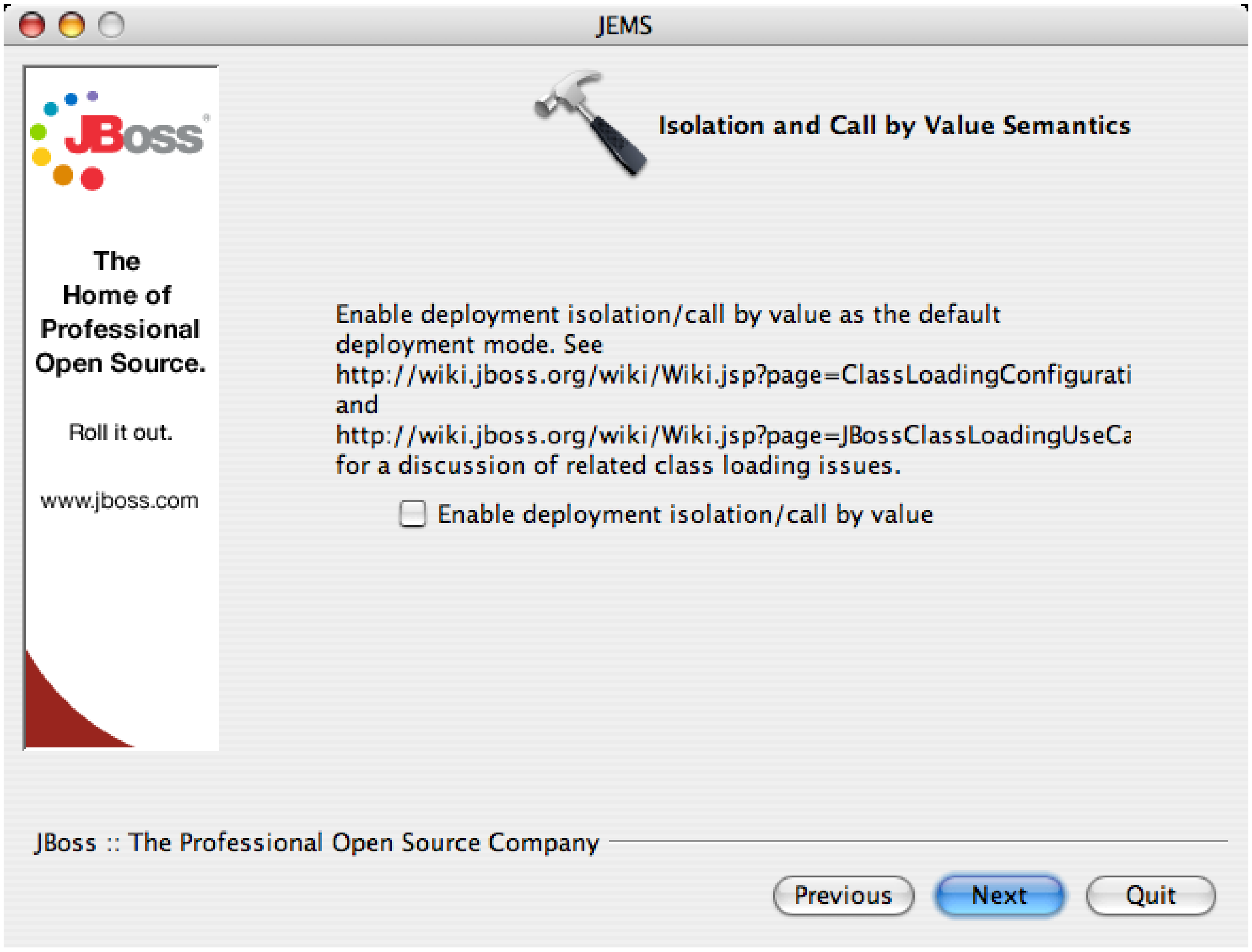
User Name:	dbadmin
Password	***********
Min Pool Size:	5
Max Pool Size:	20
Connection URL:	jdbc:JdbcProgress:T:dbserver:3305:liveappl
Driver Class:	com.progress.sql.jdbc.JdbcProgressDriver

JBoss :: The Professional Open Source Company

Figure 1.8. Configure the default datasource

Note that when installing a datasource this way, you must place the correct JDBC driver JAR file the lib directory of your server configuration. The datasource will not be usable until this is done. See Chapter 3 for more information about JBoss directory structure, including the lib directory.

The next screen allows you to enable applications isolation, completely separating the classloading space of all applications. Application isolation can be helpful in some instances, but it comes with the cost of requiring slow pass-by-value semantics for passing data between applications. In most cases, it is preferable to use loader repositories to control the sharing of classes on an application-by-application basis rather than enabling isolation for the entire server.



Isolation and Call by Value Semantics

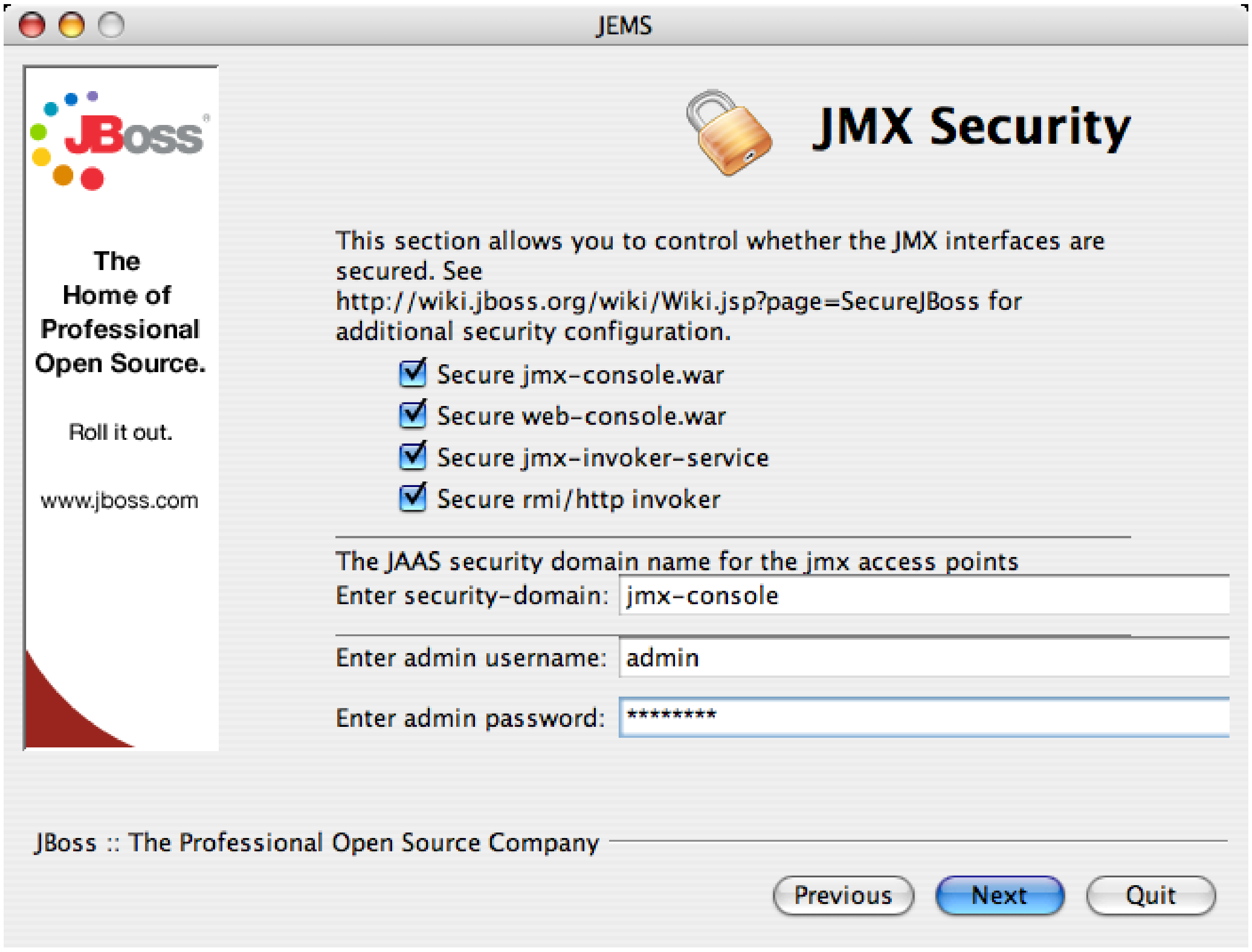
Enable deployment isolation/call by value as the default deployment mode. See http://wiki.jboss.org/wiki/Wiki.jsp?page=ClassLoadingConfigurati and http://wiki.jboss.org/wiki/Wiki.jsp?page=JBossClassLoadingUseCa for a discussion of related class loading issues.

☐ Enable deployment isolation/call by value

JBoss :: The Professional Open Source Company

Figure 1.9. Configure application isolation

When installed from a raw archive, all JBoss services are left in a developer-friendly state requiring no authentication to access most JBoss services, including administrative services. The installer gives you a chance to secure those services on the security screen, shown in Figure 1.10. It is recommended that you enable security for all services. You will be required to enter a password for the admin user. We strongly recommend not using the default password, "admin".



JMX Security

This section allows you to control whether the JMX interfaces are secured. See http://wiki.jboss.org/wiki/Wiki.jsp?page=SecureJBoss for additional security configuration.

- [✓] Secure jmx-console.war
- [✓] Secure web-console.war
- [✓] Secure jmx-invoker-service
- [✓] Secure rmi/http invoker

The JAAS security domain name for the jmx access points

Enter security-domain: jmx-console

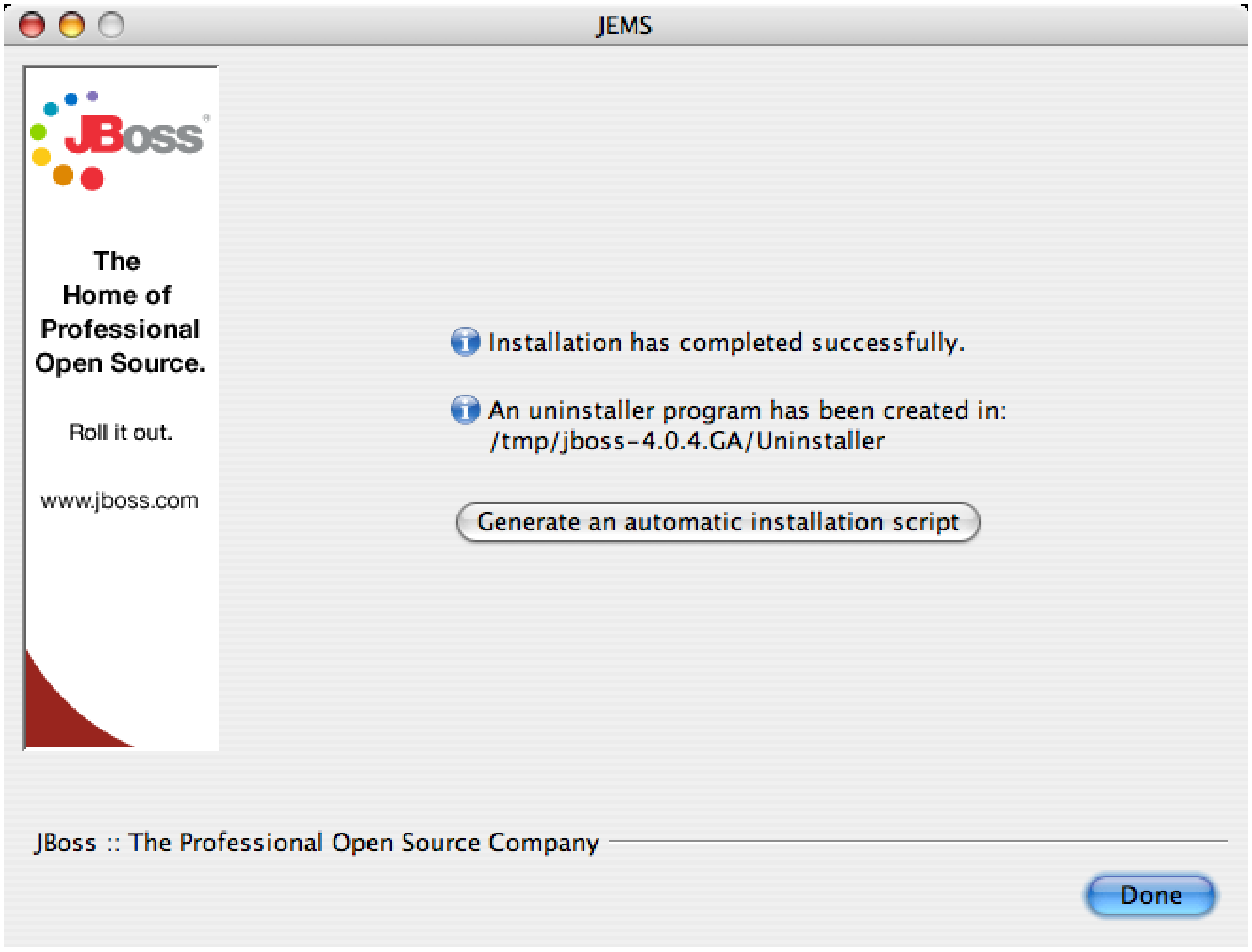
Enter admin username: admin

Enter admin password: ********

JBoss :: The Professional Open Source Company

Figure 1.10. JBoss security settings

The following screens will ask you to confirm your installation and then show you the installation progress. When your installation is done, the completion screen (Figure 1.11) gives you the option to save an installation script that can be used to recreate your installation configuration. The installation script will be covered in Chapter 2


JEMS
JBoss
The Home of Professional Open Source.
Roll it out.
www.jboss.com

Installation has completed successfully.

An uninstaller program has been created in:
/tmp/jboss-4.0.4.GA/Uninstaller

Generate an automatic installation script
JBoss :: The Professional Open Source Company
Done

Figure 1.11. The installation is completed

Your JBoss installation can be found in the directory that you specified at the beginning of the install. The installer image may contain different services than the archive distribution, depending on the type of installation performed. However, the basic structure and layout of all JBoss instances are the same. The JBoss directory structure will be explained in Chapter 3.

2

Command-line installation

Running the GUI installer requires a graphical display to run. but many servers run in headless mode with no graphical display options. In these cases, the installer can be run on a machine with a display, and the resulting JBoss installation can be copied to the server machine. Another option for installing in these environments is to use the command-line installation capabilities. The JBoss installer can be run from the command-line installing either a base server profile or recreating a previous installation using the installer script created during a previous installation.

When you want to install a specific configuration set, the installer can be run from the command line using the -installGroup and installpath options:

```
$ java -jar jems-installer-1.2.0.jar -installGroup default installpath=/opt/jboss-4.0.4
```

The -installGroup parameter is any valid installation group for the installer being used. Different JBoss installers install have different options. The options for the JEMS 1.2 installer are shown in Table 1.1. The installpath options specifies the directory you want to install JBoss into. If you do not specify the installpath option, the installer defaults to a directory named jboss-install.

This type of installation provides no ability to specify specific packages or to use the security and datasource configuration options. These options can be set when using an installer script. Script generation requires running the actual GUI interface, but the generated script can be used from the command-line . The last panel of the installer provides the option to save an installer script that preserves all of the selections made in the installer. The installation can be run by passing the installation script as a parameter to the installer.

```
$ java -jar jems-installer-1.2.0.jar installscript.xml
```

The installation script is a simple XML file that can be safely edited if further customization is needed. The most likely detail to change would be the installation directory.

3

Directory Structure

Installing the JBoss distribution creates a `jboss-4.0.4` directory that contains server start scripts, JARs, server configuration sets and working directories. You need to know your way around the distribution layout to locate JARs for compiling code, updating configurations, deploying your code, etc. Figure 3.1 illustrates the installation directory of the JBoss server.

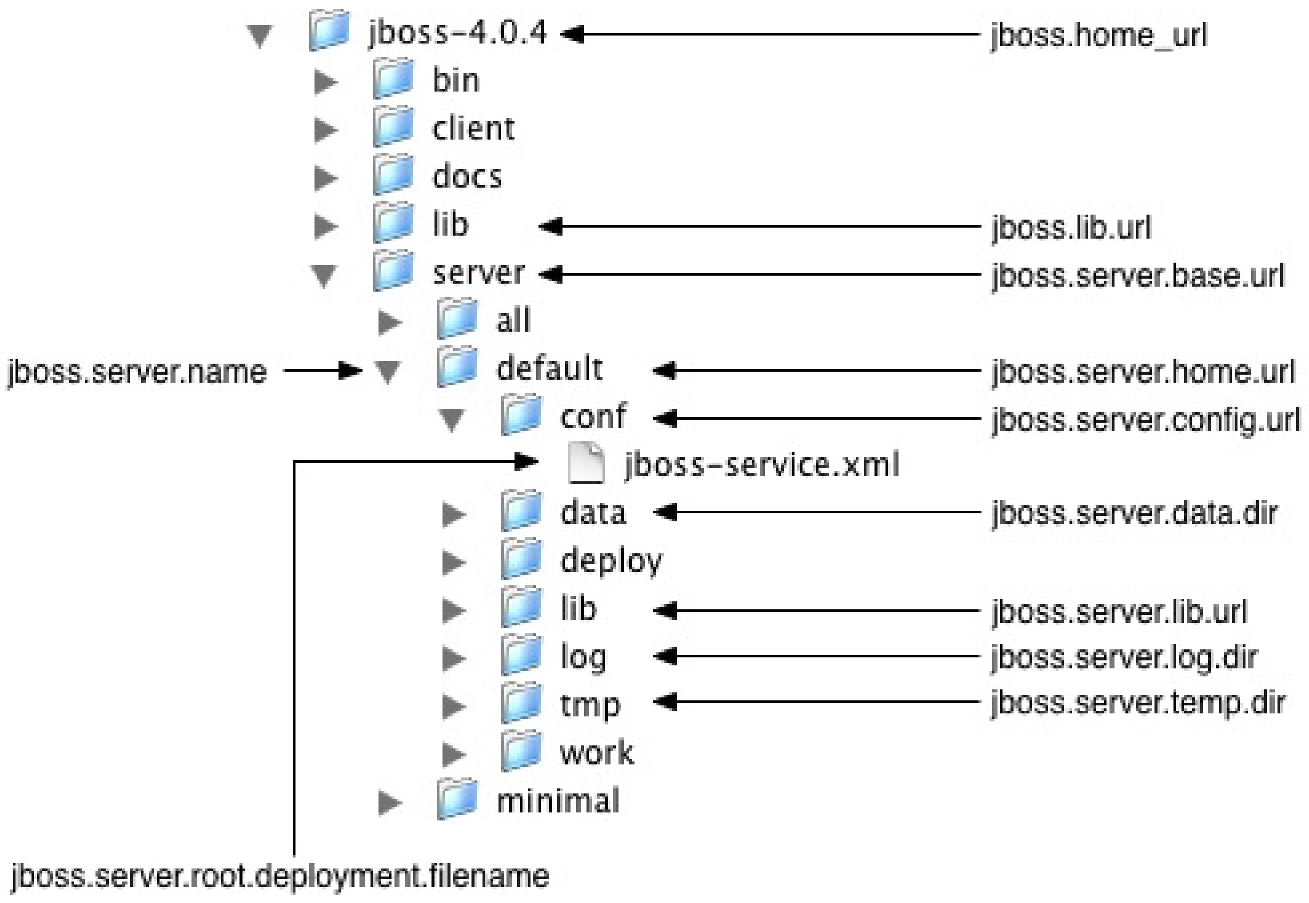


Figure 3.1. The JBoss AS directory structure

Throughout this book we refer to the top-level `jboss-4.0.4` directory as the `JBOSS_DIST` directory. In Figure 3.1, the `default` server configuration file set is shown expanded. It contains a number of subdirectories: `conf`, `data`, `deploy`, `lib`, `log`, and `tmp`. In a clean installation, only the `conf`, `deploy`, and `lib` directories will exist. Several of the locations may be overridden. For these locations, the `org.jboss.system.server.ServerConfig` interface constant and its corresponding system property string are shown. The names ending in `URL` correspond to locations that can be specified using a URL to access remote locations, for example, HTTP URLs against a web server. Table 3.1 shows the top-level directories and their function.

Table 3.1. The JBoss top-level directory structure

Directory	Description
`bin`	All the entry point JARs and start scripts included with the JBoss distribution are located in the `bin` directory.
`client`	The JARs that are required for clients that run outside of JBoss are located in the `client` directory.
`server`	The JBoss `server` configuration sets are located under the `server` directory. The default server configuration set is the `server/default` set. JBoss ships with `minimal`, `default` and `all` configuration sets. The subdirectories and key configuration files contained `default` configuration set are discussed in more detail in Chapter 4
`lib`	The `lib` directory contains startup JARs used by JBoss. Do not place your own libraries in this directory.

Table 3.2 shows the directories inside of the server configuration directory and their function.

Table 3.2. The JBoss server configuration directory structure

Directory	Description
`conf`	The `conf` directory contains the `jboss-service.xml` bootstrap descriptor file for a given server configuration. This defines the core services that are fixed for the lifetime of the server.
`data`	The `data` directory is available for use by services that want to store content in the file system.
`deploy`	The `deploy` directory is the default location the hot deployment service looks to for dynamic deployment content. This may be overridden through the `URLDeploymentScanner` URLs attribute.
`lib`	The `lib` directory is the default location for static Java libraries that should not be hot deployed. All JARs in this directory are loaded into the shared classpath at startup.
`log`	The `log` directory is the directory log files are written to. This may be overridden through the `conf/log4j.xml` configuration file.
`tmp`	The `tmp` directory is used by JBoss to store temporarily files such as unpacked deployments.

The contents of the `conf` and `deploy` directories will be shown in the following section.

4

The Default Server Configuration File Set

The `JBOSS_DIST/server` directory contains one or more configuration file sets. The `default` JBoss configuration file set is located in the `JBOSS_DIST/server/default` directory. JBoss allows you to add more than one configuration set so a server can easily be run using alternate configurations. Creating a new configuration file set typically starts with copying the `default` file set into a new directory name and then modifying the configuration files as desired. Figure 4.1 below shows the contents of the `default` configuration file set.

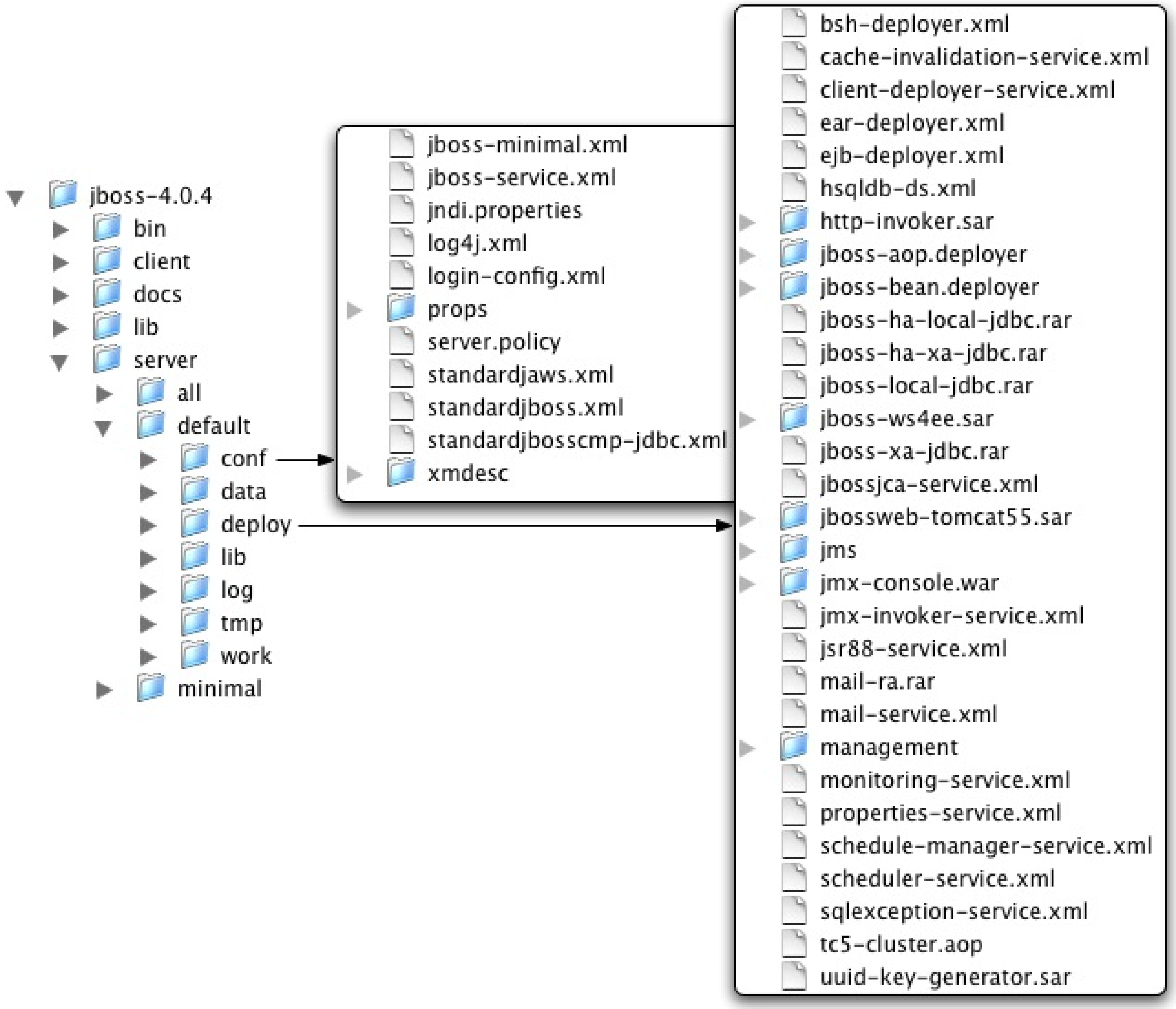


Figure 4.1. An expanded view of the default server configuration file set conf and deploy directories

The files in the `conf` directory are explained in the following section.

jboss-minimal.xml
This is a minimalist example of the `jboss-service.xml` configuration file. It is the `jboss-service.xml` file used in the `minimal` configuration file set.

jboss-service.xml
`jboss-service.xml` defines the core services configurations.

jndi.properties
The `jndi.properties` file specifies the JNDI `InitialContext` properties that are used within theNS-MYC10 JBoss server when an `InitialContext` is created using the no-argument constructor.

log4j.xml
This file configures the Apache log4j framework category priorities and appenders used by the JBoss server code.

login-config.xml
This file contains sample server side authentication configurations that are applicable when using JAVAS based security.

props/*
The `props` directory contains the users and roles property files for the `jmx-console`.

standardjaws.xml
This file provides the default configuration for the legacy EJB 1.1 CMP engine.

standardjboss.xml
This file provides the default container configurations.

standardjbosscmp-jdbc.xml
This file provides a default configuration file for the JBoss CMP engine.

xmdesc/*-mbean.xml
The `xmdesc` directory contains XMBean descriptors for several services configured in the `jboss-service.xml` file.

The following are the files in the deploy directory and their function.

bsh-deployer.xml
This file configures the bean shell deployer, which deploys bean shell scripts as JBoss services.

cache-invalidation-service.xml
This is a service that allows for custom invalidation of the EJB caches via JMS notifications. It is disabled by default.

client-deployer-service.xml
This is a service that provides support for J2EE application clients. It manages the `java:comp/env` enterprise naming context for client applications based on the `application-client.xml` descriptor.

ear-deployer.xml
The EAR deployer is the service responsible for deploying J2EE EAR files.

ejb-deployer.xml
The EJB deployer is the service responsible for deploying J2EE EJB JAR files.

hsqldb-ds.xml
`hsqldb-ds.xml` configures the Hypersonic embedded database service configuration file. It sets up the embedded database and related connection factories.

http-invoker.sar
`http-invoker.sar` contains the detached invoker that supports RMI over HTTP. It also contains the proxy bindings for accessing JNDI over HTTP.

jboss-aop.deployer,
This service configure the `AspectManagerService` and deploys JBoss AOP applications.

jboss-bean.deployer
`jboss-bean.deployer` provides the JBoss microcontainer, which deploys POJO services wrapped in `.beans` files.

jboss-ha-local-jdbc.rar
`jboss-ha-local-jdbc.rar` is an experimental version of `jboss-local-jdbc.rar` that supports datasource failover.

jboss-ha-xa-jdbc.rar
`jboss-ha-xa-jdbc.rar` is an experimental version of `jboss-xa-jdbc.rar` that supports datasource failover.

jboss-local-jdbc.rar
`jboss-local-jdbc.rar` is a JCA resource adaptor that implements the JCA `ManagedConnectionFactory` interface for JDBC drivers that support the `DataSource` interface but not JCA.

jboss-xa-jdbc.rar
`jboss-xa-jdbc.rar` is a JCA resource adaptor that implements the JCA `ManagedConnectionFactory` interface for JDBC drivers that support the `XADataSource` interface.

jbossjca-service.sar
`jbossjca-service.sar` is the application server implementation of the JCA specification. It provides the connection management facilities for integrating resource adaptors into the JBoss server.

jbossweb-tomcat55.sar
The `jbossweb-tomcat55.sar` directory provides the Tomcat 5.5 servlet engine. The SAR is unpacked rather than deployed as a JAR archive so that the tomcat configuration files can be easily edited.

jbossws14.sar
`jbossws14.sar` provides J2EE web services support.

jms/hsqldb-jdbc-state-service.xml
`hsqldb-jdbc-state-service.xml` provides JMS state management using Hypersonic.

jms/hsqldb-jdbc2-service.xml

`hsqldb-jdbc2-service.xml` configures JMS persistence and caching using Hypersonic. It also contains the `DestinationManager` MBean, which is the core service for the JMS implementation.

jms/jbossmq-destinations-service.xml

`jbossmq-destinations-service.xml` configures a number of JMS queues and topics used by the JMS unit tests.

jms/jbossmq-httpil.sar

`jbossmq-httpil.sar` provides a JMS invocation layer that allows the use of JMS over HTTP.

jms/jbossmq-service.xml

The `jbossmq-service.xml` file configures the core JBossMQ JMS service.

jms/jms-ds.xml

The `jms-ds.xml` file configures the JBossMQ JMS provider for use with the `jms-ra.rar` JCA resource adaptor.

jms/jms-ra.rar

`jms-ra.rar` is a JCA resource adaptor that implements the JCA `ManagedConnectionFactory` interface for JMS connection factories.

jms/jvm-il-service.xml

`jvm-il-service.xml` configures the in-JVM JMS transport invocation layer.

jms/uil2-service.xml

`uil2-service.xml` configures the JMS version 2 unified invocation layer. Its a fast and reliable custom socket based transport that should be used for messaging between JVMs.

jmx-console.war

The `jmx-console.war` directory provides the JMX Console. The JMX Console provides a simple web interface for managing the MBean server.

jmx-invoker-service.sar

`jmx-invoker-service.sar` is an unpacked MBean service archive that exposes a subset of the JMX `MBeanServer` interface methods as an RMI interface to enable remote access to the JMX core functionality. This is similar to the legacy `jmx-rmi-adaptor.sar`, with the difference that the transport is handled by the detached invoker architecture.

jsr-88-service.xml

`jsr-88-service.xml` provides the JSR 88 remote deployment service.

mail-ra.rar

`mail-ra.rar` is a resource adaptor that provides a JavaMail connector.

mail-service.xml

The `mail-service.xml` file is an MBean service descriptor that provides JavaMail sessions for use inside the JBoss server.

management/console-mgr.sar

`console-mgr.sar` provides the Web Console. It is a web application/applet that provide a richer view of the JMX server management data than the JMX console. You may view the console using the URL `http://localhost:8080/web-console/`.

monitoring-service.xml
The `monitoring-service.xml` file configures alert monitors like the console listener and email listener used by JMX notifications.

properties-service.xml
The `properties-service.xml` file is an MBean service descriptor that allows for customization of the JavaBeans `PropertyEditors` as well as the definition of system properties.

scheduler-service.xml, **schedule-manager-service.xml**
The `scheduler-service.xml` and `schedule-manager-service.xml` files are MBean service descriptors that provide a scheduling type of service.

sqlexception-service.xml
The `sqlexception-service.xml` file is an MBean service descriptor for the handling of vendor specific `SQLExceptions`.

uuid-key-generator.sar
The `uuid-key-generator.sar` service provides a UUID-based key generation facility.

The all configuration contains several addition services.

cluster-service.xml
This service configures clustering communication for most clustered services in JBoss.

deploy-hasingleton-service.xml
This provides the HA singleton service, allowing JBoss to manage services that must be active on only one node of a cluster.

deploy.last/farm-service.xml
`farm-service.xml` provides the farm service, which allows for cluster-wide deployment and undeployment of services.

httpha-invoker.sar
This service provides HTTP tunneling support for clustered environments.

iiop-service.xml
This provides IIOP invocation support.

juddi-service.sar
This service provides UDDI lookup services.

snmp-adaptor.sar
This is a JMX to SNMP adaptor. It allows for the mapping of JMX notifications onto SNMP traps.

tc5-cluster.sar
Provides AOP support for field-level HTTP session replication.

When installing EJB3 support, several additional EJB3 services are made available.

ejb3-interceptors-aop.xml

This service provides the AOP interceptor stack configurations for EJB3 bean types.

ejb3.deployer
This service deploys EJB3 applications into JBoss.

jboss-aop-jdk50.deployer
This is a Java 5 version of the AOP deployer. The AOP deployer configures the `AspectManagerService` and deploys JBoss AOP applications.

jbossws.sar
This services provides Java EE 5 web services support.

Finally, in the EJB3 all configuration adds two additional services:

ejb3-clustered-sfsbcache-service.xml
This services provides replication and failover for EJB3 stateful session beans.

ejb3-entity-cache-service.xml
This services provides a clustered cache for EJB3 entity beans.

5

Starting and Stopping JBoss

After you have installed the JBoss distribution, it is wise to perform a simple startup test to validate that there are no major problems with your Java VM/operating system combination. To test your installation, move to the `bin` directory and execute the `run.bat` or `run.sh` script, as appropriate for your operating system. Your output should look like the following and contain no error or exception messages:

```
$ sh run.sh
=========================================================================

  JBoss Bootstrap Environment

  JBOSS_HOME: /tmp/jboss-4.0.4.GA

  JAVA: java

  JAVA_OPTS: -server -Xms128m -Xmx512m -Dsun.rmi.dgc.client.gcInterval=3600000
             -Dsun.rmi.dgc.server.gcInterval=3600000 -Dprogram.name=run.sh

  CLASSPATH: /tmp/jboss-4.0.4.GA/bin/run.jar:/lib/tools.jar

=========================================================================

23:28:48,561 INFO  [Server] Starting JBoss (MX MicroKernel)
...
23:29:09,249 INFO  [Server] JBoss (MX MicroKernel) [4.0.4.GA (build:
  CVSTag=JBoss_4_0_4_GA date=200605151000)] Started in 20s:679ms
```

If your output is similar to this (accounting for installation directory differences), you are now be ready to use JBoss.

Using `run.sh` without any arguments starts the server using the `default` server configuration file set. To start with an alternate configuration file set, you pass in the name of the directory under `JBOSS_DIST/server` that you want to use as the value to the `-c` command line option. For example, to start with the `minimal` configuration file set you would specify:

```
$ ./run.sh -c minimal
...
23:37:41,582 INFO  [Server] JBoss (MX MicroKernel) [4.0.4.GA (build:
  CVSTag=JBoss_4_0_4_GA date=200605151000)] Started in 2s:212ms
```

The `run` script supports the following options:

```

usage: run.sh [options]
  -h, --help                    Show this help message
  -V, --version                 Show version information
  --                            Stop processing options
  -D<name>[=<value>]            Set a system property
  -d, --bootdir=<dir>           Set the boot patch directory; Must be absolute or url
  -p, --patchdir=<dir>          Set the patch directory; Must be absolute or url
  -n, --netboot=<url>         Boot from net with the given url as base
  -c, --configuration=<name>  Set the server configuration name
  -B, --bootlib=<filename>    Add an extra library to the front bootclasspath
  -L, --library=<filename>    Add an extra library to the loaders classpath
  -C, --classpath=<url>       Add an extra url to the loaders classpath
  -P, --properties=<url>      Load system properties from the given url
  -b, --host=<host or ip>     Bind address for all JBoss services
  -g, --partition=<name>      HA Partition name (default=DefaultDomain)
  -u, --udp=<ip>              UDP multicast address
  -l, --log=<log4j|jdk>       Specify the logger plugin type
```

To shutdown the server, you simply issue a Ctrl-C sequence in the console in which JBoss was started. Alternatively, you can use the `shutdown.sh` command.

```
[bin]$ ./shutdown.sh -S
```

The `shutdown` script supports the following options:

```
A JMX client to shutdown (exit or halt) a remote JBoss server.

usage: shutdown [options] <operation>

options:
  -h, --help                Show this help message (default)
  -D<name>[=<value>]        Set a system property
  --                        Stop processing options
  -s, --server=<url>        Specify the JNDI URL of the remote server
  -n, --serverName=<url>    Specify the JMX name of the ServerImpl
  -a, --adapter=<name>      Specify JNDI name of the MBeanServerConnection to use
  -u, --user=<name>         Specify the username for authentication
  -p, --password=<name>     Specify the password for authentication

operations:
  -S, --shutdown            Shutdown the server
  -e, --exit=<code>         Force the VM to exit with a status code
  -H, --halt=<code>         Force the VM to halt with a status code
```

Use of the shutdown command requires a server configuration that contains `jmx-invoker-service.xml` service, so the shutdown command cannot be used with the `minimal` configuration.