

# ***Red Hat Application Server and Developer Suite***

## **Installation Guide**



# Red Hat Application Server and Developer Suite: Installation Guide

Copyright © 2004, 2005 by Red Hat, Inc.

1801 Varsity Drive  
Raleigh NC 27606-2072 USA  
Phone: +1 919 754 3700  
Phone: 888 733 4281  
Fax: +1 919 754 3701  
PO Box 13588  
Research Triangle Park NC 27709 USA

Manual identifier:

- PDF: rhaps-rhds-EN-4-PDF-RHI (2005-06-21T16:18)
- HTML: rhaps-rhds-EN-4-HTML-RHI (2005-06-21T16:18)

Red Hat is a registered trademark and the Red Hat Shadow Man logo, RPM, and the RPM logo are trademarks of Red Hat, Inc. Linux is a registered trademark of Linus Torvalds.

Intel<sup>™</sup>, Pentium<sup>™</sup>, Itanium<sup>™</sup>, and Celeron<sup>™</sup> are registered trademarks of Intel Corporation.

EJB<sup>™</sup>, J2EE<sup>™</sup>, JCA<sup>™</sup>, JCEE<sup>™</sup>, JDBC<sup>™</sup>, JDO<sup>™</sup>, JMS<sup>™</sup>, RMI<sup>™</sup>, and Sun<sup>™</sup>, and Sun Microsystems® are registered trademarks of Sun Microsystems, Inc.

The JOnAS logo is copyright © Bruno Bellamy.

Apache is a trademark of The Apache Software Foundation.

All other trademarks and copyrights referred to are the property of their respective owners.

Copyright © 2004, 2005 by Red Hat Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

HTML, PDF, and RPM versions of the manuals are available on the Documentation CD and online at <http://www.redhat.com/docs/>.

The GPG fingerprint of the security@redhat.com key is:

CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

# Table of Contents

<b>Introduction.....</b>	<b>i</b>
1. Document Conventions.....	i
2. How to Use This Manual.....	ii
3. We Need Feedback.....	ii
<b>1. Before You Begin.....</b>	<b>1</b>
1.1. Verify Your Operating System.....	1
1.2. Verify Your System Architecture.....	1
1.3. Do You Have Enough Disk Space?.....	1
1.4. Do You Have a JDK?.....	1
1.5. If Installing from Red Hat Network.....	1
1.6. If Installing from ISO Images.....	1
<b>2. Installing from RHN or from an ISO Image.....</b>	<b>3</b>
2.1. Installing using Red Hat Network (RHN) Channels.....	3
2.2. Installing from an ISO Image.....	3
<b>3. Using the Graphical Installer.....</b>	<b>7</b>
<b>4. Packages Included in Red Hat Application Server and Red Hat Developer Suite.....</b>	<b>17</b>
<b>A. Removing Red Hat Application Server and Red Hat Developer Suite.....</b>	<b>31</b>
A.1. Removing with the Graphical Installer.....	31
A.2. Removing Using the Command Line.....	32
<b>B. Setting up Red Hat Application Server to Run with Oracle.....</b>	<b>33</b>
<b>Index.....</b>	<b>35</b>



This guide describes the system requirements for Red Hat Application Server (RHAPS) and Red Hat Developer Suite (RHDS), the packages that will be installed, how to perform the installation, and how to troubleshoot any installation problems that may occur. There is also information on how to remove RHAPS and RHDS from your server, if required.

## 1. Document Conventions

Certain words in this manual are represented in different fonts, styles, and weights. This highlighting indicates that the word is part of a specific category. The categories include the following:

Courier font

Courier font represents commands, file names and paths, and prompts.

When shown as below, it indicates computer output:

Desktop	about.html	logs	paulwesterberg.png
Mail	backupfiles	mail	reports

**bold Courier font**

Bold Courier font represents text that you are to type, such as: **service jonas start**

*italic Courier font*

Italic Courier font represents a variable, such as an installation directory: *install\_dir/bin/*

**bold font**

Bold font represents **application programs** and **text found on a graphical interface**.

When shown like this: **OK**, it indicates a button on a graphical application interface.

Additionally, the manual uses different strategies to draw your attention to pieces of information. In order of how critical the information is to you, these items are marked as follows:



### Note

Linux is case-sensitive: a rose is not a ROSE is not a rOsE.



### Tip

The directory `/usr/share/doc/` contains additional documentation for packages installed on your system.

**Important**

If you modify the DHCP configuration file, the changes will not take effect until you restart the DHCP daemon.

**Caution**

Do not perform routine tasks as root—use a regular user account unless you need to use the root account for system administration tasks.

**Warning**

Be careful to remove only the listed partitions. Removing other partitions could result in data loss or a corrupted system environment.

## 2. How to Use This Manual

This manual provides both quick and detailed installation instructions. If you are an experienced System Administrator, everything you need is in Chapter 2 *Installing from RHN or from an ISO Image*. If you require more details, start with Chapter 1 *Before You Begin* to confirm that your system can run Red Hat Application Server and Red Hat Developer Suite, then go to Chapter 2 *Installing from RHN or from an ISO Image* for detailed installation instructions.

## 3. We Need Feedback

If you have thought of a way to make this manual better, submit a bug report against the documentation component of the product Red Hat Application Server or Red Hat Developer Suite in Bugzilla at: <http://bugzilla.redhat.com/bugzilla/>

When submitting a bug report, be sure to mention the manual's identifier:

```
rhaps-rhds-EN-4-PDF-RHI (2005-06-21T16:18)
```

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

If you have a support question (for example, if you are not sure how to partition your hard drives), use the online support system by registering your product at: <http://www.redhat.com/apps/activate/>

## Before You Begin

Before you begin, ensure that your system meets the requirements listed below.

### 1.1. Verify Your Operating System

You can install Red Hat Application Server and Red Hat Developer Suite only on Red Hat Enterprise Linux 4 (RHEL4). Verify that you have this version of Red Hat Enterprise Linux running on your target hardware.

You should update your system to the latest packages available with Red Hat Network (RHN). See the Red Hat Network documentation at <http://rhn.redhat.com/help/> for details.

### 1.2. Verify Your System Architecture

Red Hat Application Server and Red Hat Developer Suite is available on x86, Itanium, and PPC architectures.

The most recent list of supported hardware can be found at: <http://hardware.redhat.com/hcl/>

### 1.3. Do You Have Enough Disk Space?

The base installation of Red Hat Application Server and Red Hat Developer Suite requires 250 MB of free disk space, in addition to the space required by your applications. Installing Tomcat adds 20 MB to that total. Before you start the installation process, ensure that your system meets this requirement.

If you are not sure that you meet this condition, or if you want to know how to create free disk space, refer to your *Red Hat Enterprise Linux Installation Guide*.

### 1.4. Do You Have a JDK?

Red Hat Application Server and Red Hat Developer Suite require that an appropriate JDK be present. The Application Server and Developer Suite were tested with a 1.4.2-level JDK. The IBM JDK and BEA JRockit JDK are available from Red Hat Network's RHEL4 Extras Channel.

Information on installing one of these JDKs is available in Section 2.1 *Installing using Red Hat Network (RHN) Channels*.

### 1.5. If Installing from Red Hat Network

Installs from Red Hat Network require that you be subscribed to the RHEL4 Extras Channel and the Red Hat Application Server channel. If you are not subscribed to these channels, see the Red Hat Network documentation at <http://www.redhat.com>.

## 1.6. If Installing from ISO Images

Red Hat Application Server and Red Hat Developer Suite is available in ISO format from Red Hat Network and the <http://ftp.redhat.com> site. However, the JDKs and PostgreSQL packages are not on the ISO image. The JDK packages are available from the RHEL4 Extras ISO image or the RHEL4 Extras RHN channel. The postgresql-server package is available from the RHEL4 Base RHN channel.

Follow steps 1 through 6 in Section 2.1 *Installing using Red Hat Network (RHN) Channels* to install the necessary packages before continuing with Section 2.2 *Installing from an ISO Image*.



## Installing from RHN or from an ISO Image

This chapter describes how to install Red Hat Application Server and Red Hat Developer Suite.

**Note**

Upgrades are supported only via RHN; run:

```
up2date
```

### 2.1. Installing using Red Hat Network (RHN) Channels

These steps describe how to use Red Hat Network Channels to install Red Hat Application Server and Developer Suite on your Red Hat Enterprise Linux 4 system. You will need to have a registered account on RHN and to have obtained access to the Red Hat Application Server channel. This is usually done as part of a subscription or evaluation process. The registered target system will need to be installed with Red Hat Enterprise Linux 4 and must have either direct access to RHN, or be a user of an RHN Proxy Server or RHN Satellite Server.

This is the installation procedure:

1. Install the target system with Red Hat Enterprise Linux 4.
2. Register the system with RHN.
3. Using your enterprise account or evaluation access, subscribe your system to the Red Hat Application Server 2.0 and Red Hat Enterprise Linux 4 Extras channels on RHN. To obtain these entitlements, contact your Red Hat account manager. When available, the packages should appear in the **Software** tab. For more information about how to use RHN, see: <https://rhn.redhat.com/help/>
4. Perform an `up2date` to ensure that the current versions of the packages are installed. Red Hat Application Server and Red Hat Developer Suite were tested with a fully-updated Red Hat Enterprise Linux 4 installation.
5. If the IBM Java JDK or BEA JRockit JDK is not installed on the system, use RHN to install your JDK of choice by running either:  

```
up2date java-1.4.2-ibm-devel
```

or  

```
up2date java-1.4.2-bea-devel
```
6. If you do not have the PostgreSQL RDBMS set up on your system with JDBC support, use RHN to install it by running:  

```
up2date postgresql-jdbc postgresql-server
```
7. Install the Red Hat Application Server. RHN will handle the cross-dependencies automatically, installing and updating software packages as needed:  

```
up2date jonas jonas-docs jonas-examples jonas-client rh-jonas-docs
```
8. Install the Red Hat Developer Suite. RHN will handle the cross-dependencies automatically, installing and updating software packages as needed:  

```
up2date eclipse-jdt eclipse-pde eclipse-cdt eclipse-rpm \  
eclipse-ve eclipse-changelog eclipse-lomboz
```

## 2.2. Installing from an ISO Image



### Note

Red Hat Application Server and Red Hat Developer Suite are available in ISO format from Red Hat Network and the <http://ftp.redhat.com/pub/redhat/linux/enterprise/4/en/RHAPS/> site.

#### 1. Obtain the Red Hat Application Server and Developer Suite:

- To obtain the Red Hat Application Server and Red Hat Developer Suite ISO from RHN:
  - a. Log into your account at: <http://rhn.redhat.com/>
  - b. Select Channels.
  - c. Select "Download Software" and download the Red Hat Application Server and Developer Suite ISO image.
- To obtain the Red Hat Application Server and Red Hat Developer Suite ISO from the <http://ftp.redhat.com> site:
  - a. Point your browser to <http://ftp.redhat.com/pub/redhat/linux/enterprise/4/en/RHAPS/>.
  - b. Select the architecture of your target machine.
  - c. From the `isos` directory, select the ISO image to start the download.

#### 2. You may install using the ISO (as root) by either directly installing the RPMs or by using the graphical installer. The graphical installer is described in detail in Chapter 3 *Using the Graphical Installer*. Refer to this section now if you are installing using the graphical installer.

#### 3. If you are directly installing the RPMs, how you proceed depends on whether you chose to burn a DVD of the ISO:

- If you chose to burn a DVD of the ISO, upon inserting it you will most likely be presented with the choice of running the graphical installer; select **No**. Should you not be prompted, run the following commands (as root) to mount the DVD:

```
mkdir /mnt/rhaps
mount /dev/cdrom /mnt/rhaps
```

- If you chose not to burn a DVD of the ISO, you will need to mount the ISO image. Do so with the following commands (as root):

```
mkdir /mnt/rhaps
mount -o ro,loop path_to_iso /mnt/rhaps
```

#### 4. Add one of the following lines to `/etc/sysconfig/rhn/sources`:

- To use `up2date` via yum repo on the disk:
 

```
yum RedHat-Application-Server-V2 file:///mnt/rhaps
```
- To use `up2date` directly on RPMs on the disk:
 

```
dir RedHat-Application-Server-V2 /mnt/rhaps/RedHat/RPMS
```

## 5. Install RHAPS:

- For the Application Server (which includes an embedded Tomcat), run:  
`up2date jonas jonas-client jonas-docs jonas-examples rh-jonas-docs`
- For the stand-alone Tomcat web container:  
`up2date tomcat5 tomcat5-webapps tomcat5-admin-webapps`

6. After the installation is complete, remove the line you added to `/etc/sysconfig/rhn/sources` in Step 4. Failure to do this will result in `up2date` warning you about `/mnt/rhps/RedHat/RPMS` not existing whenever you run `up2date` again without mounting the RHAPS V2 DVD.



## Using the Graphical Installer

If you chose to burn a DVD of the ISO, upon inserting the DVD you will most likely be presented with the choice of running the graphical installer; select “Yes”. Should you not be prompted, run the following commands (as `root`) to mount the DVD:

```
mkdir /mnt/rhps  
mount /dev/cdrom /mnt/rhps
```

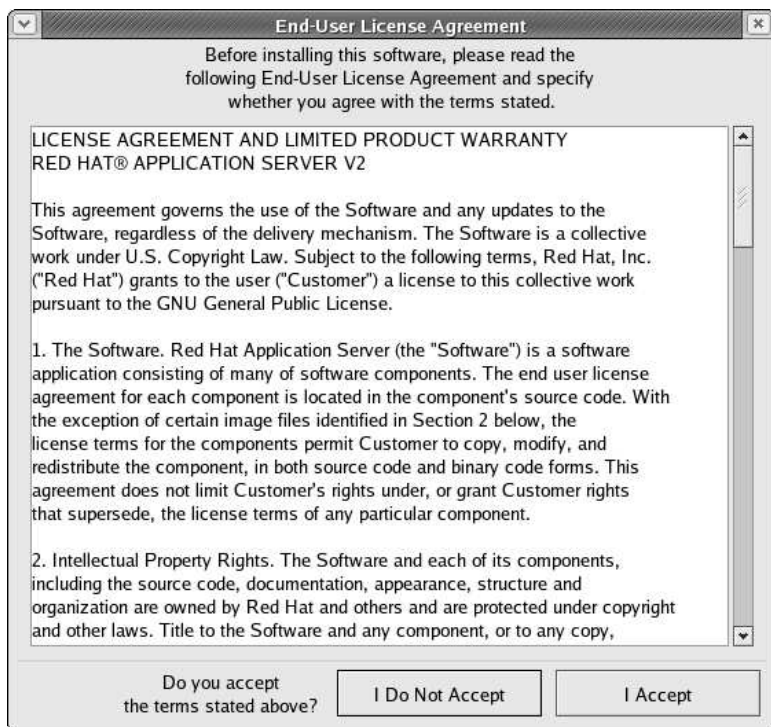
If you chose not to burn a DVD of the ISO, you will need to mount the ISO image. Do this by running the following commands (as `root`):

```
mkdir /mnt/rhps  
mount -o loop path_to_iso /mnt/rhps
```

Once the DVD or ISO image is mounted, run the following command (as `root`) to start the installer:

```
cd /mnt/rhps  
./autorun
```

After you start the installer, you are presented with the License Agreement.



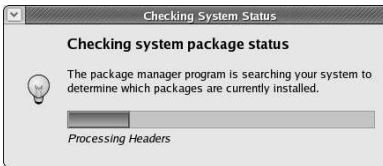
**Figure 3-1. License Agreement**

Click **I Accept** to continue.



**Figure 3-2. Package Management Introduction**

Click **Forward** to begin your installation. The installer will perform some checks to determine which packages it can install and if any are already present on your system.



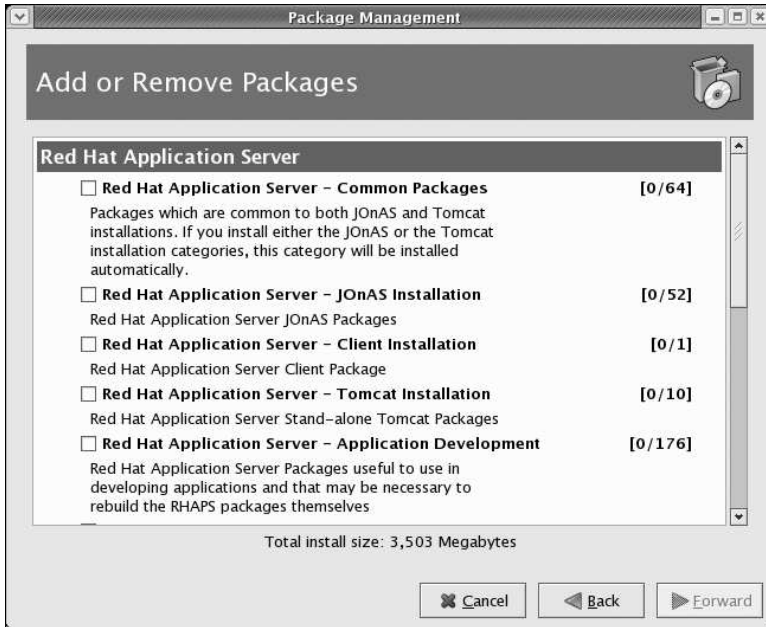
**Figure 3-3. Package Checks**

Now you can begin selecting which parts of Red Hat Application Server and Developer Suite you want to install. Packages have been grouped into categories for easy selection. The following is a brief description of the package categories:

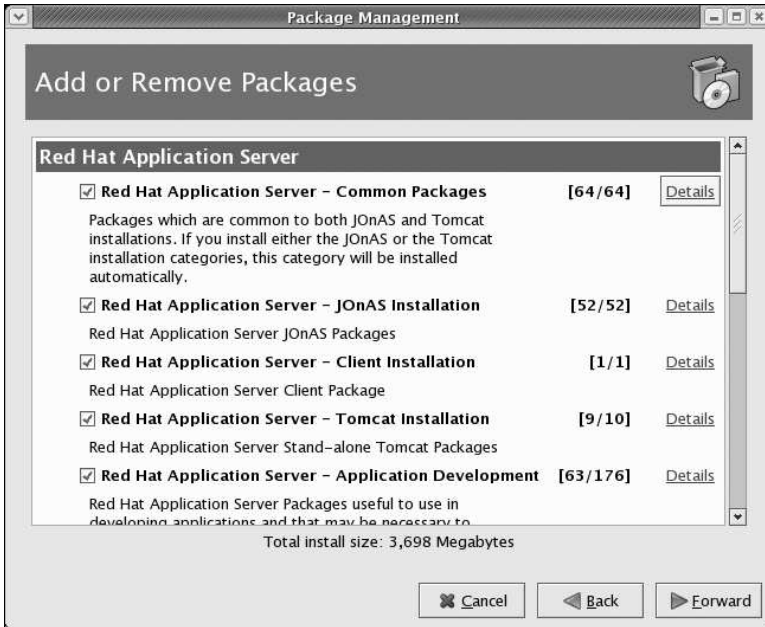
Package Category	Description
Red Hat Application Server - Common Packages	These packages are required for both JOnAS and Tomcat installations. Selecting either JOnAS or Tomcat installation will automatically install this category as well.

Package Category	Description
Red Hat Application Server - JOnAS Installation	The packages for the JOnAS installation.
Red Hat Application Server - Client Installation	The packages for the client installation.
Red Hat Application Server - Tomcat Installation (Standalone)	The packages for the standalone Tomcat installation.
Red Hat Application Server - Application Development	Packages useful for developing applications which may also be necessary to rebuild RHAPS packages.
Red Hat Application Server - Examples and Demos	These packages are examples and demonstrations for various components of Red Hat Application Server.
Red Hat Application Server - Documentation	Red Hat Application Server documentation in HTML and PDF forms.
Red Hat Developer Suite - Eclipse SDK	The packages for Eclipse Platform, Java Development Tools, Plugin Development Environment, and Documentation
Red Hat Developer Suite - CDT	The packages for C/C++ Development Tools.
Red Hat Developer Suite - GEF	The packages for the Graphical Editing Framework.
Red Hat Developer Suite - RPM Plugin	The packages for the RPM Plugin.
Red Hat Developer Suite - Visual Editor	The packages for Eclipse Visual Editor Support.
Red Hat Developer Suite - ChangeLog Plugin	The packages for the ChangeLog Plugin.
Red Hat Developer Suite - Lomboz Plugin	The packages for Application Server development tools.

**Table 3-1. Package Categories**

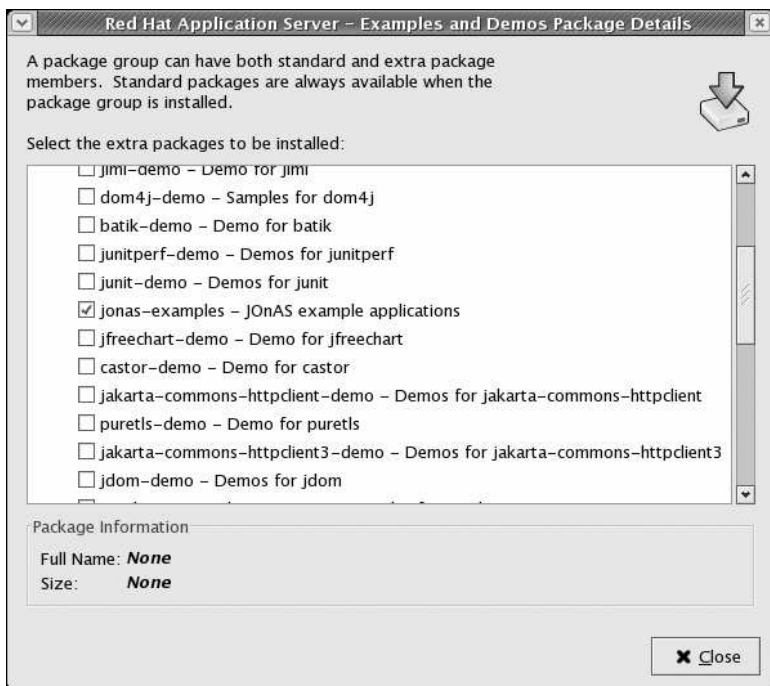
**Figure 3-4. Category Selection**

Once a package category is selected, the individual packages within that category can be viewed by clicking the **Details** button. Hold the mouse over the word **Details** to make the button appear:



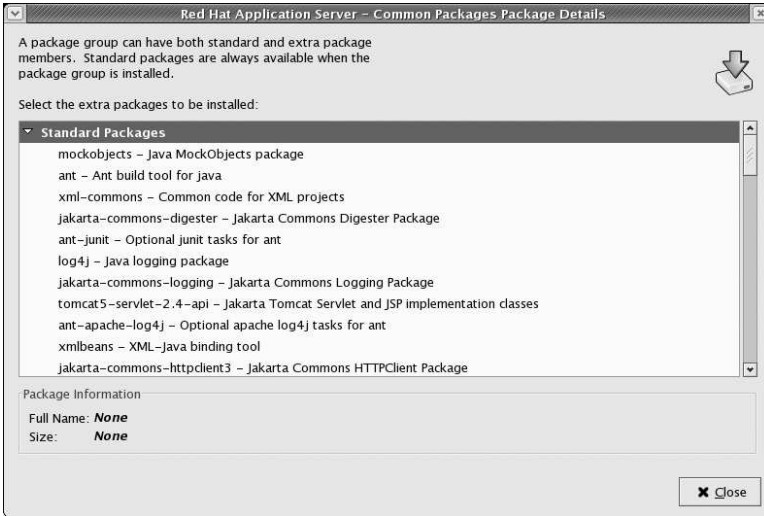
**Figure 3-5. Package Details**

For the **Red Hat Application Server - Examples and Demos** and **Documentation** categories, clicking **Details** enables you to pick and choose individual packages that you would like to install:



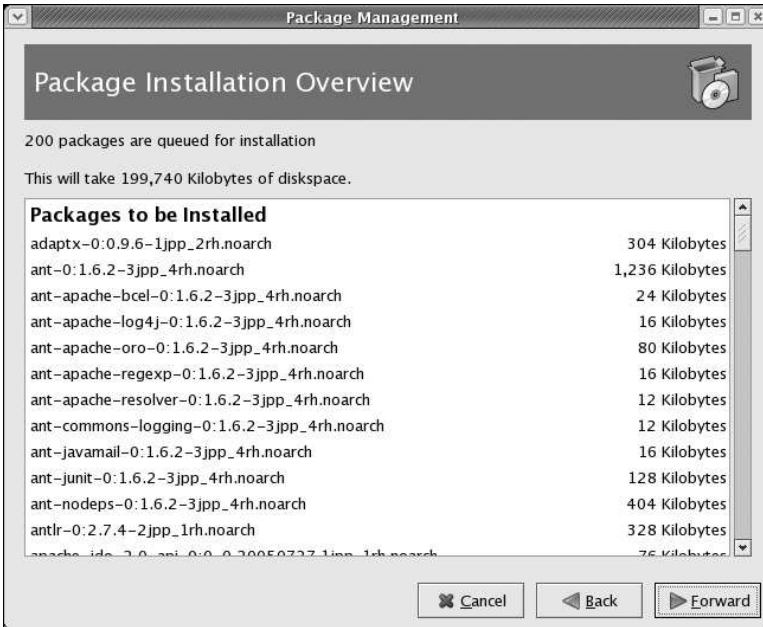
**Figure 3-6. Choosing Packages**

Other than for the Application Development, the Examples and Demos, and the Documentation categories, you cannot individually select and/or un-select packages within the category:



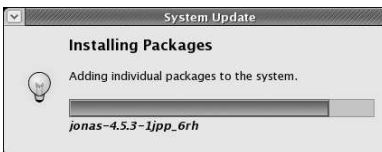
**Figure 3-7. Some Packages Occur as Groups Only**

After you have made all your selections, clicking **Forward** gives you an overview of what packages will be installed and their space requirements:



**Figure 3-8. Package Installation Overview**

During installation of the packages, the installer will display progress information:



**Figure 3-9. Package Installation Progress**

Once the installation has completed, the installer indicates that it has finished successfully:

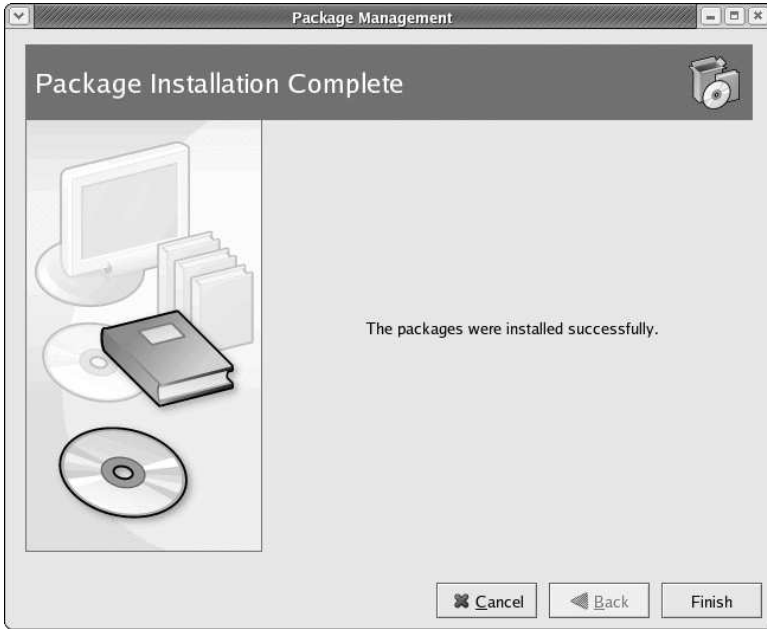


Figure 3-10. Installation Complete

## Packages Included in Red Hat Application Server and Red Hat Developer Suite

Red Hat Application Server contains Application Server Packages: JOnAS-specific Packages, JOnAS Client Packages, and stand-alone Tomcat Packages.

Red Hat Developer Suite contains Developer Suite Packages.

Package	Description
adaptx	Adaptx is an extensible Stylesheet Language (XSL) processor.
ant	Ant is a platform-independent build tool for java.
apache-jdo-2.0-api	The standard definition of the JDO API as defined by the JSR-243 standard.
antlr	ANTLR (ANother Tool for Language Recognition) is a language tool that provides a framework for constructing recognizers, compilers, and translators from grammatical descriptions containing C++ or Java actions.
asm	A code manipulation tool to implement adaptable systems.
avalon-framework	Avalon-framework provides interfaces that define relationships between commonly used application components, best-of-practice pattern enforcements, and several lightweight convenience implementations of the generic components.
avalon-logkit	Avalon-logkit provides a logging toolkit designed for secure, performance-oriented logging in applications.
axis	Apache AXIS is an implementation of the SOAP ("Simple Object Access Protocol") submission to W3C.
batik	Batik is a Java technology based toolkit for applications that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as viewing, generation or manipulation.
bcel	The Byte Code Engineering Library is intended to give users a convenient means to analyze, create, and manipulate Java class files.
bsf	Bean Scripting Framework (BSF) is a set of Java classes which provides scripting language support within Java applications, and access to Java objects and methods from scripting languages.
bsh	BeanShell is a small, free, embeddable, Java source interpreter with object scripting language features, written in Java.

Package	Description
c-jdbc	C-JDBC is a database cluster middleware that allows any Java application (standalone application, servlet or EJB container, ...) to transparently access a cluster of databases through JDB. C-JDBC instantiates the concept of RAIDb: Redundant Array of Inexpensive Databases. The database is distributed and replicated among several nodes and C-JDBC load balance the queries between these nodes.
carol	CAROL: Common Architecture for RMI ObjectWeb Layer
concurrent	Concurrent provides standardized, efficient versions of utility classes commonly encountered in concurrent Java programming.
carol-irmi	The irmi package is a JDK agnostic RMI implementation supporting pluggable interceptors and local call optimization. It uses standard JRMP stubs and uses the javax.rmi.CORBA.PortableRemoteObjectDelegate interface to plug into any JDK.
castor	Castor is a data binding framework for Java.
classpathx-mail	A platform-independent and protocol-independent framework to build mail and messaging applications.
cryptix	Cryptix 3 is a cleanroom implementation of Sun's Java Cryptography Extensions (JCE) version 1.1. It also contains the Cryptix Provider, which delivers a wide range of algorithms and support for PGP 2.x.
cryptix-asn1	A Java crypto package and asn1 implementation.
dom4j	dom4j is an Open Source XML framework for Java. dom4j enable you to read, write, navigate, create, and modify XML documents. dom4j integrates with DOM and SAX and is seamlessly integrated with full XPath support.
dtdparser	You can use this library to parse a DTD.
ews-mapper	The Mapper is an extension of the Axis file generation framework. It provides a JaxRpcMapper interface that is used to read metadata from the JAXRPC Mapping File. The data read from that mapping file will be used to generate adequate Java classes (class names, methods names and signature, etc.) and correct WSDD.
fastrmic	A Java RMI stub/skeleton compiler that generates bytecode using the ASM library from ObjectWeb. It supports both version 1.1 and version 1.2 of Java RMI, but does not have support for IIOP.
fop	FOP is a print formatter driven by XSL formatting objects. It is a Java application that reads a formatting object tree and then turns it into a PDF document. The formatting object tree can be in the form of an XML document or it can be passed in memory as a DOM Document or SAX event.

<b>Package</b>	<b>Description</b>
forehead	Forehead is a very small framework to assist in controlling the run-time ClassLoader hierarchy of Java applications.
fractal	Fractal is a general software composition framework that supports component-based programming, including components (type) definition, configuration, composition and administration.
geronimo-specs	The J2EE-Specifications for Apache's ASF-licensed J2EE server project.
gnu-crypto	GNU Crypto provides cryptographic primitives and tools for the Java programming language.
gnu.getopt	The GNU Java getopt classes support short and long argument parsing compatible with the version of GNU getopt in glibc 2.0.6. There is also a programmer's interface.
gnu.regexp	Java NFA regular expression engine implementation
gvf	The Graph Visualization Framework is a set of Java 2 packages that can serve as a foundation for applications that manipulate or display graph structures.
howl-logger	Provides features required by the ObjectWeb JOTM project, with a public API that is generally usable by any Transaction Manager.
hsqldb	Hsqldb Database Engine
ht2html	This is the <a href="http://www.python.org">www.python.org</a> Web site generator.
httpunit	HttpUnit emulates the relevant portions of browser behavior, including form submission, JavaScript, basic http authentication, cookies and automatic page redirection, and allows Java test code to examine returned pages either as text, an XML DOM, or containers of forms, tables, and links.
ishmael	Ishmael is an open source implementation of JSR-88 which is the Deployment API for J2EE
isorelax	The ISO RELAX project has public interfaces that are useful for applications that support RELAX Coreas well as applications that are schema-language neutral.
jacorb	Free Java implementation of OMG's CORBA standard
jakarta-commons-beanutils	Jakarta Commons BeanUtils Package. The scope of this package is to create a package of Java utility methods for accessing and modifying the properties of arbitrary JavaBeans.

Package	Description
<code>jakarta-commons-betwixt</code>	The Betwixt library provides an XML introspection mechanism for mapping beans to XML in a flexible way. It is implemented using <code>XMLIntrospector</code> and <code>XMLBeanInfo</code> classes that are similar to the standard <code>Introspector</code> and <code>BeanInfo</code> from the Java Beans specification. Betwixt provides a way of turning beans into XML as well as automatically generating digester rules in a way that can be customized on a per-type manner in the same way that the <code>BeanInfo</code> mechanism can be used to customize the default introspection on a Java object.
<code>jakarta-commons-cli</code>	Jakarta Commons CLI, a Command Line Interface for Java.
<code>jakarta-commons-codec</code>	Commons Codec provides definitive implementations of commonly used encoders and decoders.
<code>jakarta-commons-collections</code>	Utilities for handling Java collections; these extend or augment the Java Collections Framework.
<code>jakarta-commons-daemon</code>	Jakarta Commons Daemon Package
<code>jakarta-commons-dbcp</code>	Jakarta Commons DataBase Pooling Package
<code>jakarta-commons-digester</code>	XML-to-Java-object mapping utility commonly used for parsing XML configuration files.
<code>jakarta-commons-discovery</code>	Discovers implementations for pluggable interfaces.
<code>jakarta-commons-el</code>	Implementation for <code>javax.servlet.jsp.el</code> .
<code>jakarta-commons-fileupload</code>	Utilities for uploading files.
<code>jakarta-commons-grant</code>	Commons Grant makes it easier to use Jakarta Ant in an embedded environment.
<code>jakarta-commons-graph</code>	Jakarta Graph is a toolkit for managing graphs and graph based data structures.
<code>jakarta-commons-httpclient</code>	The Jakarta Commons HTTP Client component provides an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations. Designed for extension while providing robust support for the base HTTP protocol, the HTTP Client component will be of interest to anyone building HTTP-aware client applications such as web browsers, web service clients, or systems that leverage or extend the HTTP protocol for distributed communication.
<code>jakarta-commons-httpclient3</code>	Provides an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.
<code>jakarta-commons-io</code>	Commons.IO is a package of Java utility classes for the <code>java.io</code> hierarchy.
<code>jakarta-commons-jelly</code>	Jelly is a Java- and XML-based scripting engine.

<b>Package</b>	<b>Description</b>
<code>jakarta-commons-jexl</code>	Jexl is an expression language engine that implements an extended version of the Expression Language of the JSTL. It is designed for easy embedding in applications and frameworks.
<code>jakarta-commons-lang</code>	Common set of utility classes that provide extra functionality for classes in <code>java.lang</code> .
<code>jakarta-commons-launcher</code>	Cross platform Java application launcher.
<code>jakarta-commons-logging</code>	Logging utilities: wrappers around a variety of logging API implementations.
<code>jakarta-commons-modeler</code>	Mechanisms to create Model MBeans compatible with the Java Management Extensions (JMX) specification.
<code>jakarta-commons-net</code>	This is an Internet-protocol-suite Java library that supports Finger, Whois, TFTP, Telnet, POP3, FTP, NNTP, SMTP, and some miscellaneous protocols such as Time and Echo. As well, there is BSD R command support. The purpose of the library is to provide fundamental protocol access, not higher-level abstractions.
<code>jakarta-commons-pool</code>	An object (instance) pooling package distributed under the ASF license.
<code>jakarta-commons-validator</code>	A framework to define validation methods and rules in XML files with support for internationalization.
<code>jakarta-taglibs-standard</code>	An open-source implementation of the JSP Standard Tag Library.
<code>java_cup</code>	A Java source interpreter.
<code>javacc</code>	Java Compiler Compiler (JavaCC) is the most popular parser generator for use with Java applications. A parser generator is a tool that reads a grammar specification and converts it to a Java program that can recognize matches to the grammar. In addition to the parser generator itself, JavaCC provides other standard capabilities related to parser generation, such as tree building (via a tool called JJTree included with JavaCC), actions, debugging, etc.
<code>jaxen</code>	Jaxen is an XPath engine written in Java to work against a variety of XML-based object models, such as DOM, dom4j and JDOM, together with Java Beans.
<code>jcommon</code>	This is a collection of classes used by Object Refinery Projects (for example, jfreechart).
<code>jdepend</code>	JDepend traverses a set of Java class and source file directories and generates design quality metrics for each Java package. JDepend enables you to automatically measure the quality of a design in terms of its extensibility, reusability, and maintainability to effectively manage and control package dependencies.
<code>jdom</code>	Java alternative to DOM and SAX.
<code>jfreechart</code>	This is a free Java class library for generating charts.

Package	Description
jgroups	Toolkit for reliable multicast communication.
jimi	Jimi is a class library for managing images. Jimi's range of supported formats includes GIF, JPEG, TIFF, PNG, PICT, Photoshop, BMP, Targa, ICO, CUR, Sunraster, XBM, XPM, and PCX, although some of these formats do not have complete support for all features.
jing	Jing implements: RELAX NG 1.0 Specification, RELAX NG Compact Syntax, and parts of RELAX NG DTD Compatibility, specifically checking of ID/IDREF/IDREFS. Jing also has experimental support for schema languages other than RELAX NG; specifically W3C XML Schema (based on Xerces-J); Schematron; Namespace Routing Language.
jlex	JLex is a Lexical Analyzer Generator for Java.
jline	JLine is a java library for reading and editing user input in console applications. It features tab-completion, command history, password masking, customizable keybindings, and pass-through handlers to use to chain to other console applications.
jonas	An open-source application server.
jonas-client	JOnAS "fat client" support.
jonas-docs	The jonas-docs package contains web application versions of the JOnAS User Guide and of the tomcat documentation.
jonas-examples	Example web applications.
jonathan-core	A Distributed Object Platform (DOP) written entirely in Java.
jonathan-jeremie	A Distributed Object Platform (DOP) written entirely in Java.
joram	Java Open Reliable Asynchronous Messaging.
jorm	JORM (Java Object Repository Mapping) is an adaptable persistence service. It can be used to offer various personalities, such as one compliant with the CMP EJB specification (TM), another with the OMG PSS specification or another with the JDO (Java Data Objects) specification (TM). JORM provides object persistency through different secondary storage supports, such as files, relational databases or object-oriented databases.
jorm-rdb-adapter	JORM (Java Object Repository Mapping) is an adaptable persistence service that provides object persistency through files, relational databases or object-oriented databases. This package provides relational database support to JORM.
jotm	JOTM : A Java Open Transaction Manager
jrefactory	A variety of refactoring and pretty printing tools.

<b>Package</b>	<b>Description</b>
jsch	JSch enables you to connect to an sshd server and use port forwarding, X11 forwarding, file transfer, etc. You can integrate its functionality into your own Java programs.
jtidy	JTidy is a Java port of HTML Tidy, an HTML syntax checker and pretty printer. Like HTML Tidy, JTidy can be used as a tool for cleaning up malformed and faulty HTML. In addition, JTidy provides a DOM parser for real-world HTML.
juddi	jUDDI (pronounced “Judy”) is an open source Java implementation of the Universal Description, Discovery, and Integration (UDDI) specification for Web Services.
junit	Java regression test package. JUnit is a regression testing framework used by developer’s who implement unit tests in Java.
junitdoclet-jdk14	JUnitDoclet generates skeletons of TestCases based on your application source code. It also supports the reorganization of tests during refactoring.
junitperf	JUnitPerf is a collection of JUnit test decorators used to measure the performance and scalability of functionality contained within existing JUnit tests.
jython	Jython is an implementation of the object-oriented language Python seamlessly integrated with the Java platform. Jython is complementary to Java and is especially suited for embedded scripting, interactive experimentation, and rapid application development. You can freely mix the two languages during development and in shipped products.
jzlib	The zlib is a free, general-purpose, legally unencumbered, lossless, data-compression library for use on virtually any computer hardware and operating system.
ldapjdk	The Mozilla LDAP Java SDK.
libreadline-java	Java wrapper for the GNU-readline library.
log4j	Java logging package. Log4j is a tool to help the programmer output log statements to a variety of output targets.
lucene	Jakarta Lucene is a high-performance, full-featured, text-search engine written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.
maven	Maven is a Java project management and project comprehension tool. Maven is based on the concept of a project object model (POM) in that all the artifacts produced by Maven are a result of consulting a well defined model for your project. Builds, documentation, source metrics, and source cross-references are all controlled by your POM.

Package	Description
medor	MEDOR (Middleware Enabling Distributed Object Requests) allows the expression, optimization and evaluation of queries on heterogeneous distributed objects. Projection to relational databases, including complex mapping, is managed through integration with JORM.
medor-expression	MEDOR (Middleware Enabling Distributed Object Requests) allows the expression, optimization and evaluation of queries on heterogeneous distributed objects. Projection to relational databases, including complex mapping, is managed through integration with JORM.
mockobjects	Java MockObjects package
mod_jk-ap20	Tomcat mod_jk connector for Apache 2.0.x.
monolog	An API of monitoring and logging.
msv-strict	The Sun Multi-Schema XML Validator (MSV) is a Java technology tool to validate XML documents against several kinds of XML schemata. It supports RELAX NG, RELAX Namespace, RELAX Core, TREX, XML DTDs, a subset of XML Schema Part 1, and JAXP masquerading.
mx4j	OpenJMX is an open source implementation of the Java(TM) Management Extensions (JMX).
nanoxml-lite	Lite version of nanoxml
nekohtml	NekoHTML is a simple HTML scanner and tag balancer that enables application programmers to parse HTML documents and access the information using standard XML interfaces. The parser can scan HTML files and "fix up" many common mistakes that human (and computer) authors make in writing HTML documents. NekoHTML adds missing parent elements; automatically closes elements with optional end tags; and can handle mismatched inline element tags.
netcomponents	This is an Internet-protocol-suite Java library that supports Finger, Whois, TFTP, Telnet, POP3, FTP, NNTP, SMTP, and some miscellaneous protocols such as Time and Echo. As well, there is BSD R command support. The purpose of the library is to provide fundamental protocol access, not higher-level abstractions.
objectweb-anttask	ObjectWeb Ant task
objectweb-deploysched	The ObjectWeb scheduling framework.
objectweb-emb	Enterprise Media Beans provide a framework to integrate rich media data (such as audio, video, or images) into applications based on EJB Entity Beans within the J2EE application development model.
octopus	Octopus is a Java-based Extraction, Transformation, and Loading (ETL) tool. It may connect to any JDBC data sources and perform transformations defined in an XML file.

<b>Package</b>	<b>Description</b>
oldjdom	Java alternative to DOM and SAX
oldkilim	A generic configuration framework for Java
oldrhino	Rhino is an open-source implementation of JavaScript written entirely in Java. It is typically embedded into Java applications to provide scripting to end users.
openorb	OpenORB is a CORBA Object Request Broker fully developed in Java. It fully complies with the CORBA 2.4.2 specification and provides a lot of features, services and extensions.
opensaml	OpenSAML is a set of open source Java and C++ libraries that implement the SAML 1.0 and 1.1 specifications.
oro	Full regular expressions API. The Jakarta-ORO Java classes are a set of text-processing Java classes that provide Perl5 compatible regular expressions, AWK-like regular expressions, glob expressions, and utility classes for performing substitutions, splits, filtering filenames, etc.
p6spy	Database statement interceptor for Java
perl-SOAP-Lite	
puretls	
perseus-cache	Perseus provides various components for managing persistency. The cache component provides caching of java objects.
perseus-concurrency	Perseus provides various components for managing persistency. The concurrency manager component manages concurrent accesses on resources by task.
perseus-dependency	Perseus provides various components for managing persistency.
perseus-distribution	Perseus provides various components for managing persistency.
perseus-fos	Perseus provides various components for managing persistency.
perseus-persistence	Perseus provides various components for managing persistency.
perseus-pool	Perseus provides various components for managing persistency. The pool component provides pooling of java objects.
regex	Simple regular expressions API.
relaxngDatatype	
rh-jonas-docs	JOnAS documentation PDF and release notes.
rhino	Rhino is an open-source implementation of JavaScript written entirely in Java. It is typically embedded into Java applications to provide scripting to end users.

Package	Description
saxon	Java XSLT processor: a collection of tools for processing XML documents.
servletapi3	The source code for the implementation classes of the Java Servlet and JSP APIs (packages javax.servlet).
speedo	Speedo is an open source implementation of the JDO 1.0.1 specification.
stax-bea	The Streaming API for XML (StAX) is a Java API for parsing and writing XML easily and efficiently.
struts	Web application framework.
tanukiwrapper	Java Service Wrapper
tomcat5	Apache Servlet/JSP Engine that conforms to the Servlet 2.4 and JSP 2.0 specifications.
tomcat5-admin-webapps	Administration web application for Tomcat.
tomcat5-servlet-2.4-api	Java servlet and JSP implementation classes.
tomcat5-webapps	Web application for Tomcat.
tribe	Tribe is a Java-based group communication library. It is based on reliable point-to-point FIFO communication channels (basically TCP) and targets high-performance cluster environments.
velocity	Java-based template engine
velocity-dvsl	DVSL (Declarative Velocity Style Language) is a tool modeled after XSLT. It is intended for general XML transformations using the Velocity Template Language for the transformations. The key differences are that it incorporates easy access to Java objects and it allows you to use the Velocity template language's features for expressing the transformation templates.
werken.xpath	XPath implementation using JDOM
werkz	Werkz is a light-weight goal-oriented process framework. Certain tasks, such as organizing the build-chain of a complex development project, can be easily modelled as a graph of goals. werkz is a framework that allows modelling of a complex graph of goals and an engine that efficiently and correctly attempts to satisfy requests. As an abstract framework, werkz imposes no particular domain semantics or execution model; it simply manages the dependencies between goals and invokes application-specific code in the correct order.
ws-fx-addressing	Apache Addressing is an implementation of the Web Services Addressing (WS-Addressing) on top of Apache Axis (the next-generation SOAP).

Package	Description
ws-jaxme	This is a Java/XML binding compiler that takes as input a schema description (in most cases an XML schema, but it can be a DTD, a RelaxNG schema, a Java class inspected via reflection, or a database schema). The output is a set of Java classes. ws-jaxme can also convert the Java bean back into the original XML document.
ws-scout	Apache Scout is an implementation of the JSR 93 (JAXR).
wsabi4j2ee-api	wsabi4j2ee-api is a set of standard APIs to interact with web services deployed on various application services. Currently it only offers read-only access to web service information on a set of application servers, it eventually aims to provide the ability to CRUD (create, read, update, delete) and manage web services on various application servers.
wsdl4j	Web Services Description Language Toolkit for Java.
wss4j	Apache WSS4J is an implementation of the OASIS Web Services Security (WS-Security) from OASIS Web Services Security TC. WSS4J is a primarily a Java library that can be used to sign and verify SOAP Messages with WS-Security information. WSS4J will use Apache Axis and Apache XML-Security projects and will be interoperable with JAX-RPC based server/clients and .NET server/clients.
xalan-j2	Xalan is an XSLT processor for transforming XML documents into HTML, text, or other XML document types. It implements the W3C Recommendations for XSL Transformations (XSLT) and the XML Path Language (XPath).
xdoclet	XDoclet Attribute Orientated Programming Framework
xerces-j2	Java XML parser. Xerces 2 is a fully conforming XML Schema processor. This version of Xerces introduces the Xerces Native Interface (XNI), a complete framework for building parser components and configurations that is extremely modular and easy to program.
xercesjarv	This driver allows you to control the XML Schema validation engine of Xerces-2 through JARV, a vendor-neutral, open-source interface for validation engines.
xjavadoc	The XJavaDoc engine
xml-commons	Common code for XML projects.
xml-commons-resolver	Resolver subproject of xml-commons; catalog-based entity and URI resolution.
xml-security	The XML Security project provides implementation of security standards for XML. Currently the focus is on the W3C standards XML-Signature Syntax and Processing and XML Encryption Syntax and Processing. A Java library and a C++ library are available.

Package	Description
xmlbeans	XML-Java binding tool
xmldb-api	The API interfaces for xmldb. A DriverManager implementation is also provided.
xmlunit	XMLUnit extends JUnit to simplify unit testing of XML. It compares a control XML document to a test document or the result of a transformation, validates documents against a DTD, and compares the results of XPath expressions.
xpp2	XML Pull Parser 2 (XPP2) is an incremental XML parser.
xsdlib	Sun XML Datatypes Library, Sun's Java technology implementation of W3C's XML Schema Part 2 ( <a href="http://www.w3.org/TR/xmlschema-2/">http://www.w3.org/TR/xmlschema-2/</a> ), is intended for use with applications that incorporate XML Schema Part 2. This preview version implements the recommendation version ( <a href="http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/">http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/</a> ) of the W3C XML Schema Part 2 Datatype.

Table 4-1. Red Hat Application Server Packages

Package	Description
eclipse-branding	Red Hat branding plug-in for Eclipse that contains Red Hat-specific information.
eclipse-cdt	The eclipse-cdt package contains Eclipse features and plugins that are useful for C and C++ development.
eclipse-changelog	The eclipse-changelog package contains Eclipse features and plugins that are useful for ChangeLog maintenance. This package provides debug information for package eclipse-changelog.
eclipse-ecj	Eclipse compiler for Java.
eclipse-emf	The eclipse-emf package provides the EMF plugins for eclipse. EMF is a modeling framework and code generation facility for building tools and other applications based on a structured data model.
eclipse-gef	The eclipse-rpm package contains Eclipse features and plugins that comprise the Graphical Editor Framework for Eclipse.
eclipse-jdt	Eclipse Java development tools.
eclipse-lomboz	The eclipse-lomboz package contains the Lomboz Eclipse plugin, a toolkit for web and J2EE (TM) applications development.
eclipse-platform	Eclipse platform common files. In addition to this, you will need the startup scripts and a UI package (GTK2) to be able to run the IDE.
eclipse-pde	Eclipse PDE.

Package	Description
eclipse-rpm	The eclipse-rpm package contains an Eclipse plugin for building RPMs.
libswt3-gtk2	SWT Library for GTK2.

Table 4-2. Red Hat Developer Suite SDK Packages





# Removing Red Hat Application Server and Red Hat Developer Suite

Red Hat Application Server and Red Hat Developer Suite can be removed using either the graphical installer or directly using the `rpm` command.

## A.1. Removing with the Graphical Installer

1. Mount the ISO as described in Section 2.2 *Installing from an ISO Image*.
2. If you are not presented with the option of running the graphical installer, run the following commands (as root):

```
cd /mnt/rhps
./autorun
```

The graphical installer will run as described in Chapter 3 *Using the Graphical Installer*. Instead of selecting package categories or individual packages to install, un-select installed portions of Red Hat Application Server and Red Hat Developer Suite that you would like to uninstall. When you have selected packages that you wish to remove, you are presented with a confirmation dialog listing these packages.

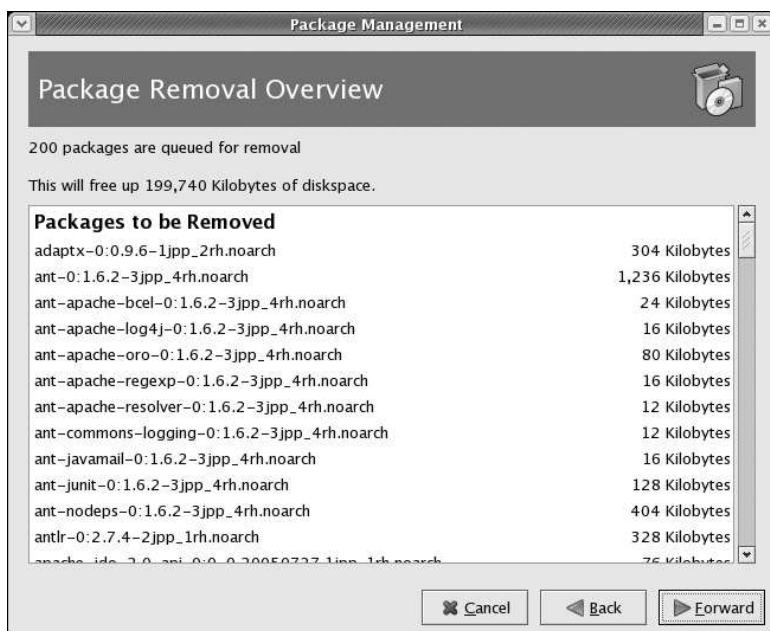
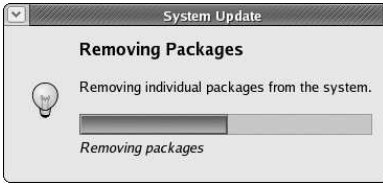


Figure A-1. Package Removal Overview

Clicking **Forward** begins the package removal. The installer displays its progress as it uninstalls packages.



**Figure A-2. Package Removal Progress**

Once the removal has completed, the installer indicates that it has finished successfully:



**Figure A-3. Package Removal Complete**

## A.2. Removing Using the Command Line

Check the Package Listings for Red Hat Application Server Red Hat Developer Suite and remove the RPMs listed there that are on your system by issuing the command:

```
rpm -e packagenames
```

## Setting up Red Hat Application Server to Run with Oracle

Red Hat Application Server comes preconfigured to use PostgreSQL as the backend. To configure Red Hat Application Server to run with Oracle:

1. Edit `$JONAS_ROOT/conf/Oracle1.properties` to fill in the appropriate values for the Oracle installation. Update the hostname, port, and SID for the `datasource.url` entry, as well as the `datasource.username` and `datasource.password`.

For example, for an Oracle installation on a server named `to-rhps1`, your properties file should have the following entries:

```
datasource.url jdbc:oracle:thin:@to-rhps1:1521:orcl
datasource.classname oracle.jdbc.driver.OracleDriver
datasource.username jonas
datasource.password jonas
```

Note that `oracle_ds` is the name that you will have to use in your beans deployment descriptors to reference this `datasource`.

2. Assuming `JONAS_ROOT` is set to the JONAS installation directory (`/usr/share/jonas` by default), create the `Oracle1_DM.rar` by running the following command as root:

```
su - jonas -s /bin/sh -c "cd $JONAS_ROOT/rars/autoload;
    $JONAS_ROOT/bin/unix/RAConfig -j jdbc_1 -dm -p $JONAS_ROOT/conf/Oracle1
    $JONAS_ROOT/rars/autoload/JONAS_jdbcDM Oracle1_DM"
```

The jndi name to use to access this source would then be `jdbc_1`. You should also ensure that "resource" is among the services listed in the `jonas.services` list in your `jonas.properties` file. If it is not, add it and restart the server.

3. Make sure you have the Oracle JDBC installed, then update `$JONAS_ROOT/bin/unix/config_env` to change the value for `ORACLE_CLASSES` to point to your JDBC driver. For example:

```
ORACLE_CLASSES=/usr/share/java/oracle-jdbc-9.0.2.0.0.jar
```

4. Deploy the resource adapter (it will be deployed automatically when the server is restarted):

```
jonas admin -a Oracle1_DM.rar
```

Optionally, you can also use the 'dbm' Service instead. Note, however, that this method is deprecated.

1. Edit your `$JONAS_ROOT/conf/Oracle1.properties` as above.
2. To use the 'dbm' service, you need to modify your `jonas.properties` file. Assuming you have called the configuration file `Oracle1.properties`, you must define a line such as:

```
jonas.service.dbm.datasources Oracle1
```

3. Ensure that "dbm" is among the services listed in the `jonas.services` list in your `jonas.properties` file. If it is not, add it and restart JONAS so that the changes become effective.



# Index

## D

disk space requirements, 1

## H

hardware requirements, 1

## I

installation

- detailed instructions, 3

- removing, 31

ISO image installation

- detailed instructions, 4

- quick instructions, 2

## P

packages in RHAPS and RHDS, 17

## R

removing RHAPS and RHDS, 31

RHN installation

- detailed instructions, 3

- quick instructions, 1

## S

system prerequisites, 1

