

Sun Cobalt Java Developer Kit Guide (v1.3)

i. Introduction

The Sun Cobalt Java Developer Kit is a set of tools to help developers start developing Java-centric applications on the Sun Cobalt appliance Platform. The kit includes the standard J2SE (Java 2 Standard Edition) development environment and the Tomcat JSP (JavaServer Pages) / servlet engine. Supported platforms include the RaQ 3, RaQ 4, RaQ XTR, and Qube 3 server appliances and installation is a click away via the server's web administration interface. For server appliances that ship with the Interbase database, the Interclient JDBC database driver has also been included in the kit.

ii. Installation instructions

The Java Developer Kit is distributed as a package file (.pkg file) that is uploaded to the Sun Cobalt server via either the administrator's "Install Software" page on the RaQ 3 and 4 or the BluelinQ page on the RaQ XTR and Qube 3. This software requires a minimum of 128MB of RAM be installed on the server appliance, and the following patch levels:

- RaQ 3 – Requires OS Update 4.0 installed
- RaQ 4 – Requires OS Update 1.0 installed
- RaQ XTR – No update needed
- Qube 3 – Requires OS Update 1.0 installed

Important information for the Qube 3 and RaQ XTR

There is a limitation with the manual install feature of the Qube 3 and RaQ XTR where .pkg's greater than 8M cannot be uploaded. The Java Developer Kit is larger than this limit. One solution is to put the file in the machine's web root and have it download the .pkg from itself. Please follow the following process to install on a Qube 3 or RaQ XTR using this technique:

- Download the .pkg to your local machine from either the Sun Cobalt website or CD
- Use an FTP client to FTP to the RaQ XTR or Qube 3
- Log in as admin
- On RaQ, cd to /home/sites/home/web, and on Qube cd to /home/groups/home/web Put the .pkg file in that location
- Go to the BluelinQ "New Software" page, choose "Install Manually"
- In the URL field, type http://localhost/[Name of the .pkg file]
- Click "Prepare" to start installation

iii. JSP and servlet operation

Once the Java Developer Kit is installed on the Sun Cobalt server, JSP and servlet functionality is enabled on the web server. Instructions for disabling the feature are given later in this document. JSP and servlets are executed via the Tomcat engine. This piece of software is a plug in to the web server already installed on the server. Once the Java Developer Kit is installed, the web server is told to send certain requests to the Tomcat engine for processing. Tomcat requires an execution environment known as a “context” be setup before JSPs and servlets can be executed properly. The Java Developer Kit sets contexts up for you automatically; there is both a default context and you can add new contexts by dropping Java .war (Web archive) files into web content directories.

Deploying JSPs and servlets on the Qube

The default Tomcat context on the Qube is the /home/groups/home/web/ directory. When the Java Developer Kit was installed, a new directory called WEB-INF was created in this location. This directory contains some configuration information for Tomcat and also a directory where servlet classes are placed for deployment. JSP pages can just be dropped in the /home/groups/home/web directory and the web server will know to pass them off to Tomcat for handling as long as they end in the .jsp extension. For example, if you had a file named test.jsp, you would place it (via FTP or file sharing) in:

```
/home/groups/home/web/test.jsp
```

The dynamic page is accessed via the URL:

[http://\[hostname\]/test.jsp](http://[hostname]/test.jsp)

The first access to this page will take a little time while the Tomcat engine compiles the page, but subsequent accesses will be fast.

Deploying a servlet on the Qube is as simple as putting the servlet class file (the output of compiling a servlet source Java file) in:

```
/home/groups/home/web/WEB-INF/classes
```

For example, if you had a file named test.class you would place it in:

```
/home/groups/home/web/WEB-INF/classes/test.class
```

The servlet is executed via the URL:

[http://\[hostname\]/servlet/test](http://[hostname]/servlet/test)

All accesses to this page are fast because no compilation has to take place. If you need to have a source file for a Java servlet, see the section below on compiling Java servlets.

JSP and servlets can also be deployed via a Java .war file. A .war file is a convenient way to deploy a web application. See the section below on deploying web applications via .war files for more information.

Deploying JSPs and servlets on the RaQ

The RaQ has the ability to host multiple domains. The Java Developer Kit will create a default Tomcat context for each site on your Sun Cobalt server in the web directory for the site:

```
/home/sites/[site name]/web
```

A new directory called WEB-INF is created in this location. This directory contains some configuration information for Tomcat and also a directory where servlet classes are placed for deployment. JSP pages can just be dropped in:

```
/home/sites/[site name]/web
```

The web server will know to pass them off to Tomcat for handling as long as they end in the .jsp extension. For example, if you had a file named test.jsp, you would place it (via FTP) in:

```
/home/sites/[site name]/web/test.jsp
```

The dynamic page is accessed via the URL:

[http://\[site name\]/test.jsp](http://[site name]/test.jsp)

The first access to this page will take a little time while the Tomcat engine compiles the page, but subsequent accesses will be fast.

Deploying a servlet on the RaQ is as simple as putting the servlet class file (the output of compiling a servlet source Java file) in:

```
/home/sites/[site name]/web/WEB-INF/classes
```

For example, if you had a file named test.class you would place it in:

```
/home/sites/[site name]/web/WEB-INF/classes/test.class
```

The servlet is executed via the URL:

[http://\[site_name\]/servlet/test](http://[site_name]/servlet/test)

All accesses to this page are fast because no compilation has to take place. If you need to have a source file for a Java servlet, see the section below on compiling Java servlets.

JSP and servlets can also be deployed via a Java .war file. A .war file is a convenient way to deploy a web application. See the section below on deploying web applications via .war files for more information.

iv. Deploying .war files

v. Compiling servlet sources

vi. Compiling servlet sources