

Building Web Applications with EclipseRT

Virgo Tutorial Guide

This guide shows you all the steps in the Virgo section of the tutorial.

Table of Contents

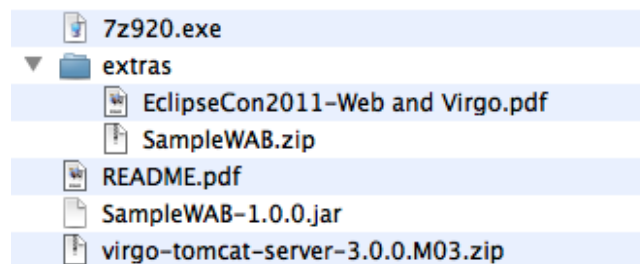
A. Get set up.....	2
B. Deploy a web app.....	4
C. Explore a web app using the admin console.....	6
D. Explore a web app using JMX.....	8
E. Configure application trace.....	9
F. Configure Tomcat.....	10
G. Configure a specific context.....	11
H. Further information.....	12

A. Get set up

1. You'll need Java 6. If you don't have it, please try downloading it. If you can't get it, follow the instructions in this guide some other time.
2. Get one of the memory sticks being passed around. Unless you're looking at someone else's display, you already have it. If you still have a memory stick at the end of the tutorial, you are welcome to keep it.



3. Copy everything (except any directories with names beginning with “.”) from the memory stick to your computer.



This includes:

- a) 7z920.exe – 7zip extractor for Windows users (see step 4).
 - b) README.pdf – this guide
 - c) SampleWAB-1.0.0.jar – the sample web app
 - d) virgo-tomcat-server-3.0.0.M03.zip – the latest and greatest milestone of Virgo Tomcat Server, including documentation.
 - e) extras directory – this contains some SDKs that will be used in the non-Virgo sections of the tutorial and, for completeness, the few slides for the Virgo section and the source of the sample web app.
4. Unzip the virgo-tomcat-server file
Avoid spaces in the directory path you unzip to.
Windows Users: use the 7zip extractor provided on the memory stick rather than the built in zip extractor.
Windows Users: Due to problems with long file paths on Windows, unzip virgo-tomcat-server at a root of a drive, e.g. in C:\.
 5. Starting the virgo-tomcat-server

Use a terminal window, change directory to the unzipped virgo-tomcat-server, change directory to the bin subdirectory, and then...

For Unix/Linux/Mac

```
[15:40:59 ~/Working/virgo-tomcat-server-3.0.0.M03]: cd bin/  
[15:41:01 ~/Working/virgo-tomcat-server-3.0.0.M03/bin]: ./startup.sh  
[2011-03-16 15:41:07.417] startup-tracker <KE0001I> Kernel starting.  
[2011-03-16 15:41:09.956] startup-tracker <KE0002I> Kernel started.
```

For Windows

```
C:\virgo-tomcat-server-3.0.0.M03\bin>startup.bat  
[2011-03-16 16:26:49.237] startup-tracker <KE0001I> Kernel starting.  
[2011-03-16 16:26:53.596] startup-tracker <KE0002I> Kernel started.
```

6. User region ready

As Virgo starts up various messages will go by announcing what is happening. The first one to notice is 'Kernel Started', this mean the Virgo kernel is up and it will now start up the user region. The next one to note is 'User region ready'. This means Virgo has started including any user defined artifacts to be started in the user region.

```
[2011-03-16 13:22:55.903] sync Event Dispatcher Thread <UR0001I> User region ready.
```

At this point Virgo will start looking for applications to deploy and there are some already available and the messages that follow are these applications being deployed.

B. Deploy a web app

1. Deploy using the pickup directory

Copy the sample web application to the `pickup` directory within the `virgo-tomcat-server`.

You should see similar output to indicate the application has been deployed.

```
[2011-03-16 16:55:57.173] fs-watcher      <HD0001I> Hot deployer processing 'CREATED' event for file 'SampleWAB-1.0.0.jar'.
[2011-03-16 16:55:57.211] fs-watcher      <DE0000I> Installing bundle 'SampleWAB' version '1.0.0'.
[2011-03-16 16:55:57.333] fs-watcher      <DE0001I> Installed bundle 'SampleWAB' version '1.0.0'.
[2011-03-16 16:55:57.376] fs-watcher      <DE0004I> Starting bundle 'SampleWAB' version '1.0.0'.
[2011-03-16 16:55:57.379] start-signalling-2 <WE0000I> Starting web bundle 'SampleWAB' version '1.0.0' with context path '/samplewab'.
[2011-03-16 16:55:57.487] start-signalling-2 <WE0001I> Started web bundle 'SampleWAB' version '1.0.0' with context path '/samplewab'.
[2011-03-16 16:55:57.488] start-signalling-2 <DE0005I> Started bundle 'SampleWAB' version '1.0.0'.
```

2. Undeploy using the pickup directory

Simply remove the sample application from the `pickup` directory.

You should see similar output to indicate the application has been undeployed.

```
[2011-03-16 16:57:01.410] fs-watcher      <HD0001I> Hot deployer processing 'DELETED' event for file 'SampleWAB-1.0.0.jar'.
[2011-03-16 16:57:01.413] fs-watcher      <WE0002I> Stopping web bundle 'SampleWAB' version '1.0.0' with context path '/samplewab'.
[2011-03-16 16:57:01.499] fs-watcher      <WE0003I> Stopped web bundle 'SampleWAB' version '1.0.0' with context path '/samplewab'.
[2011-03-16 16:57:01.501] fs-watcher      <DE0010I> Stopping bundle 'SampleWAB' version '1.0.0'.
[2011-03-16 16:57:01.504] fs-watcher      <DE0011I> Stopped bundle 'SampleWAB' version '1.0.0'.
[2011-03-16 16:57:01.506] fs-watcher      <DE0013I> Uninstalling bundle 'SampleWAB' version '1.0.0'.
[2011-03-16 16:57:01.514] fs-watcher      <DE0014I> Uninstalled bundle 'SampleWAB' version '1.0.0'.
```

3. The Splash screen app

You may have noticed some other applications in the `pickup` directory already including the splash and admin screens. Open your favourite browser and go to <http://localhost:8080>

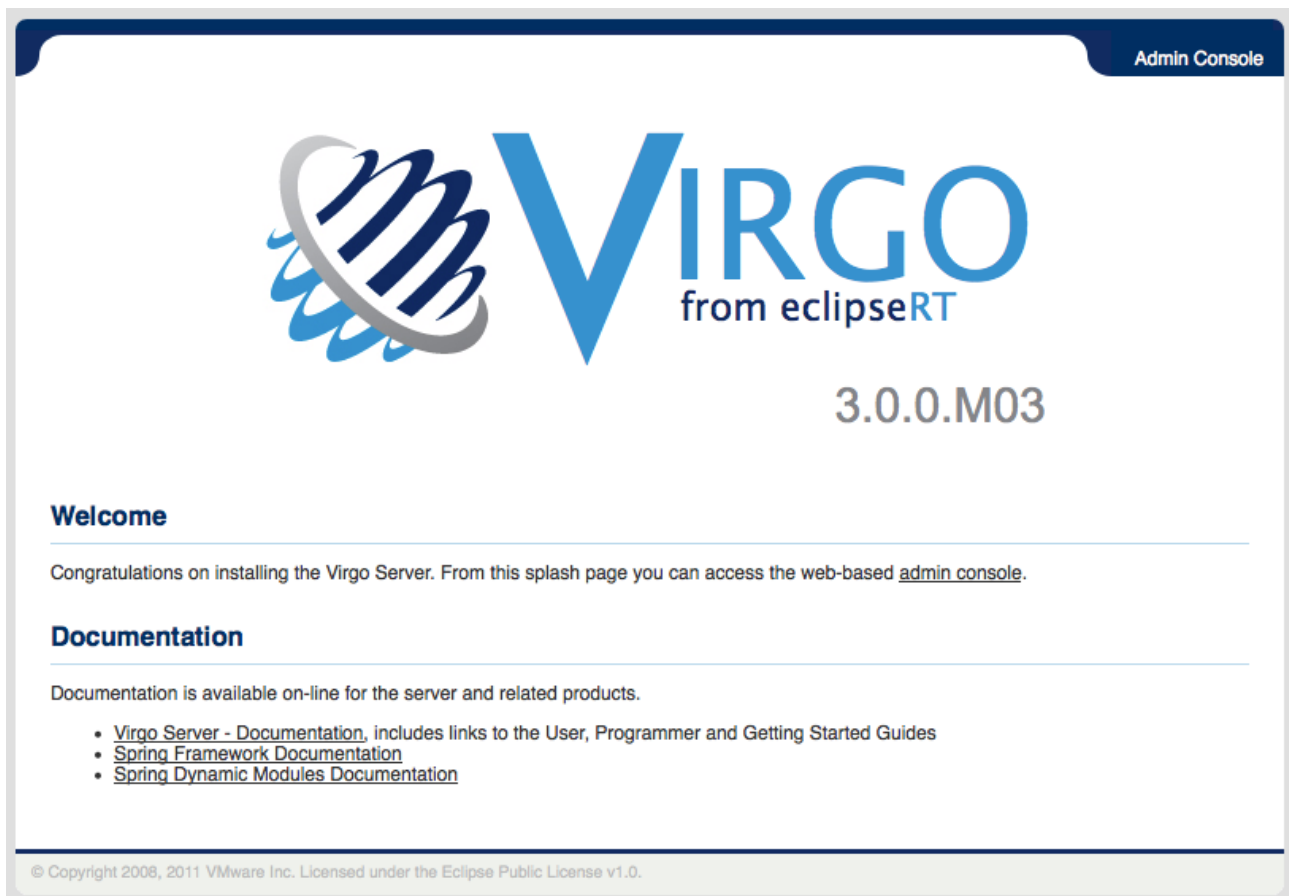
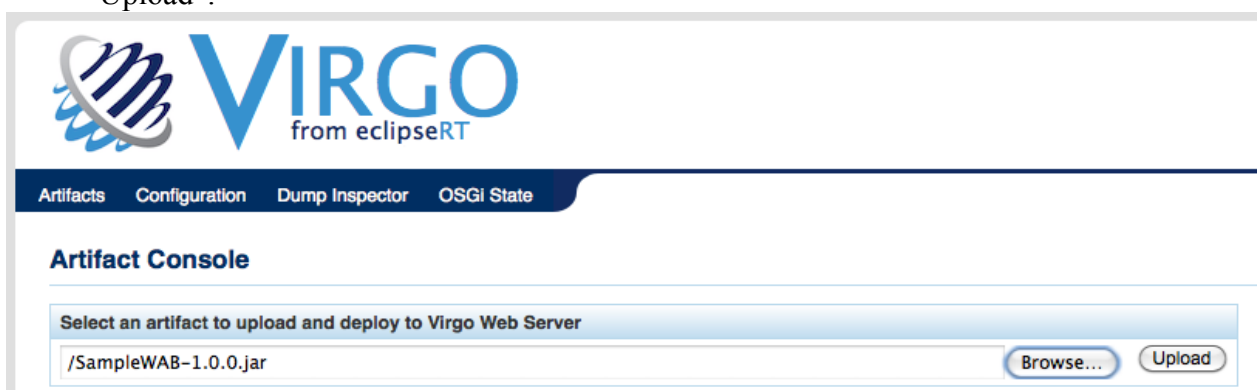


Illustration 1: Splash Screen

The splash screen shows you the version of Virgo you are running and provides a link to another application in the `pickup` directory: the Admin Console. Click it! You will require an id and password, the defaults are 'admin' and 'springsource'.

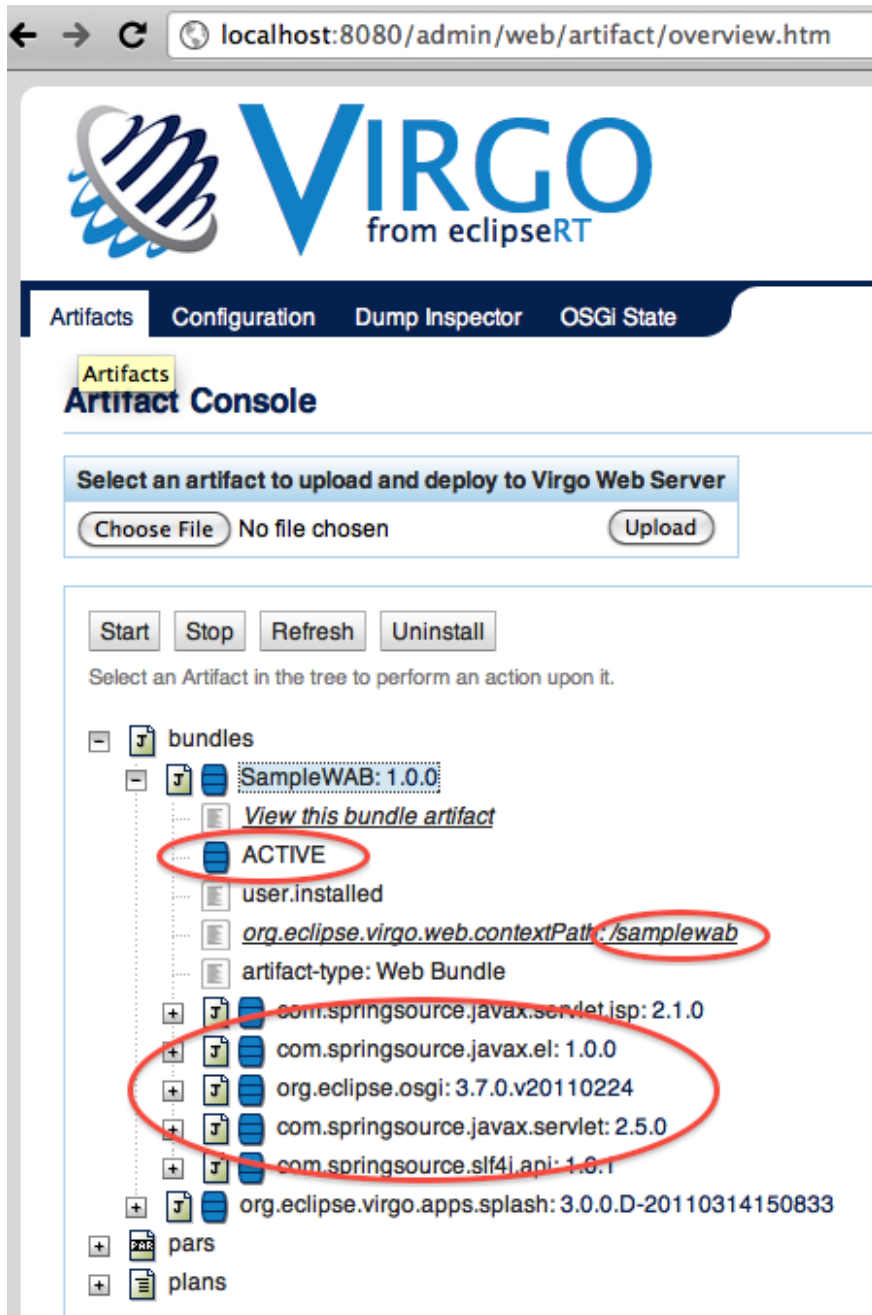
4. Deploy using the Admin console

Click on the "[Artifacts](#)" tab on the far left. Then select the sample application jar file by clicking "Browse...", "Choose file...", or similar, depending on your browser, and then click "Upload".



C. Explore a web app using the admin console

1. Open the admin console at <http://localhost:8080/admin>
2. Click on "[Artifacts](#)"
3. Expand the "+" sign next to "bundles"
4. Expand the "+" sign next to "SampleWAB: 1.0.0"
5. Observe the state of the web app, its context path, and its dependencies



6. Now click on the link "[View this bundle artifact](#)"
7. Expand the "Published Services" section
8. Observe the ServletContext service published in the service registry (the service id you see will probably not be 201):

← → ↻ localhost:8080/admin/web/state/bundle.htm?name=SampleWAB&version=1.0.0



Artifacts Configuration Dump Inspector OSGi State

[Bundles Overview](#) [Services Overview](#) Viewing state 'Live': Live ↕ Go

Viewing bundle 'SampleWAB: 1.0.0'

Bundle Symbolic name	SampleWAB
Bundle Version	1.0.0
Bundle ID	104
Hosts/Fragments	NA
Spring powered	No
State	Active
Bundle Location	org.eclipse.virgo.region.user@file:/Users/glynnormington/eclipsecon/virgo-tomcat-se

- ▶ Imported Packages (228)
- ▶ Exported Packages (0)
- ▶ Required Bundles (0)
- ▶ Consumed Services (0)
- ▼ Published Services (1)
 - ▼ Service 201 - ObjectClass javax.servlet.ServletContext

Property	Value
osgi.web.version	1.0.0
osgi.web.contextpath	/samplewab
osgi.web.symbolicname	SampleWAB
service.id	201
objectClass	javax.servlet.ServletContext

9. **Exercise:** Go back to the artifacts view and try stopping the sample web app and see what difference this makes in the artifacts view and when viewing the bundle artifact.

D. Explore a web app using JMX

1. Run `jconsole` from a terminal window
2. Connect to the `org.eclipse.virgo.osgi.launcher.Launcher` process
3. Select the Mbeans view.
4. Navigate through `org.eclipse.virgo.kernel->Model->bundle->SampleWAB`
5. Observe the state of the web app, its context path, and its dependencies. Note: to display the value field of attributes such as Dependents, double-click the Value field. Double-click again to close the view.

The screenshot displays the Java Monitoring & Management Console. The left sidebar shows a tree view of the MBean hierarchy, with `org.eclipse.virgo.kernel>Model>bundle>SampleWAB` selected. The main pane shows the 'Attribute values' tab for the `SampleWAB` MBean. The 'Dependents' attribute is expanded, showing a list of dependent bundles. The 'Properties' section shows the `key` property with the value `org.eclipse.virgo.web.contextPath/samplewab`. The 'State' property is `ACTIVE`, the 'Type' is `bundle`, and the 'Version' is `1.0.0`.

Name	Value
org.eclipse.virgo.kernel:type=Model,artifact-type=bundle,name=com.springsource.javax.el,version=1.0.0	
org.eclipse.virgo.kernel:type=Model,artifact-type=bundle,name=com.springsource.slf4j.api,version=1.6.1	
org.eclipse.virgo.kernel:type=Model,artifact-type=bundle,name=com.springsource.javax.servlet.jsp,version=2.1.0	
org.eclipse.virgo.kernel:type=Model,artifact-type=bundle,name=com.springsource.javax.servlet,version=2.5.0	
org.eclipse.virgo.kernel:type=Model,artifact-type=bundle,name=org.eclipse.osgi,version=3.7.0.v20110224	

Name	Value
key	org.eclipse.virgo.web.contextPath/samplewab
value	

State: ACTIVE
Type: bundle
Version: 1.0.0

E. Configure application trace

1. Copy and paste the following to `config/serviceability.xml` as a child of the `<configuration>` element:

```
<logger level="DEBUG" additivity="false" name="sample">
  <appender-ref ref="SIFTED_LOG_FILE" />
  <appender-ref ref="LOG_FILE" />
</logger>
```

2. Restart Virgo: the admin console does allow you to do this , so type `Ctrl-C` in the terminal in which Virgo is running and then start it again
3. Drive the sample web app at <http://localhost:8080/samplewab/>
4. A new directory `serviceability/logs/SampleWAB-1.0.0` will have been created. Look in `serviceability/logs/SampleWAB-1.0.0/log.log` to see debug logging output:

```
[2011...] http-8080-5 sample.SampleServlet enter SampleServlet.doGet
[2011...] http-8080-5 sample.SampleServlet exit SampleServlet.doGet
```

F. Configure Tomcat

1. Edit config/tomcat-server.xml
2. For example, add the following request dumper valve configuration as a child of the `<Engine>` element:

```
<Valve className="org.apache.catalina.valves.RequestDumperValve"/>
```

3. Restart Virgo the admin console does allow you to do this , so type Ctrl-C in the terminal in which Virgo is running and then start it again
4. Drive the sample web app at <http://localhost:8080/samplewab/>
5. Look in serviceability/logs/virgo-server/log.log to see requests and responses:

```
[2011...] http-8080-1 System.err ... o.a.c.valves.RequestDumperValve invoke
[2011...] http-8080-1 System.err INFO: REQUEST URI=/samplewab/
[2011...] http-8080-1 System.err ... o.a.c.valves.RequestDumperValve invoke
[2011...] http-8080-1 System.err INFO: authType=null
etc.
```

G. Configure a specific context

1. Remove the request dumper valve from `config/tomcat-server.xml`
2. Create a directory `config/Catalina/localhost`
3. Create a file in that directory called “`samplewab.xml`” with the following contents:

```
<Context>
```

```
    <Valve className="org.apache.catalina.valves.RequestDumperValve" />
```

```
</Context>
```

4. Restart Virgo: the admin console does allow you to do this , so type Ctrl-C in the terminal in which Virgo is running and then start it again
5. Drive the sample web app at <http://localhost:8080/samplewab/>
6. Look in `serviceability/logs/SampleWAB-1.0.0/log.log` to see requests and responses in the application trace log.

H. Further information

For Virgo user forum and other information, please go to <http://www.eclipse.org/virgo>.

For Tomcat configuration information, please go to <http://tomcat.apache.org/tomcat-6.0-doc/config/index.html>.