Learning the Wonders

An introduction to creating great web-applications with Project Wonder

Deployment

by

Markus Ruggiero

Table of Contents

Deployment

Deployment

1. Deployment

The following text is an excerpt from the Book Learning The Wonders. It is part of the chapter about deployment.

I have installed and deployed Wonder applications on Mac OS X. Here is a detailed description on how to do it on a virgin Mac OS X Mountain Lion Server.

The example covers Mac OS X Mountain Lion Server plus PostgreSQL.

1.1. Setting up the environment

We start with an out-of-the-box installation of Mac OS X Mountain Lion Client (currently 10.8.3), then download the corresponding Server.app.

Run Server.app. During its first run it will initialize various things and prepare different directories and settings. Don’t do anything, just run it, let it do whatever it is doing, then quit Server.app.

This step kind of converts your client installation into a server. You can continue to use your Mac as if it were a client install (which it still is basically).

However there are some differences between client and server. Most notably is the setup of Apache. On a client installation you have Apache work off /etc/apache2 and use the config files from there. The default htdoc-directory is /Library/WebServer/Documents. When you convert your client to become a server, Apache will use the configuration from /Library/Server/Web/Config/apache2 and have its default htdocs in /Library/Server/Web/Data/Sites/Default. Just be aware of these different paths.

1.1.1. Using the Apple provided PostgreSQL

There is one more configuration Server.app will set up: PostgreSQL. Server uses PostgreSQL for its own purposes. You can of course work with the same PostgreSQL installation.

As there is no PostgreSQL option in Server.app you must either enable a service that uses PostgreSQL or start PostgresSQL it manually. This is easiest done from the command line.

```
$ sudo serveradmin status postgres
postgres:state = "STOPPED"
$ sudo serveradmin start postgres
postgres:state = "RUNNING"
```

And while we are there, we start the webserver as well. This however could also be accomplished from within Servera.app.

```
$ sudo serveradmin status web
web:state = "STOPPED"
$ sudo serveradmin start web
web:state = "RUNNING"
```

1.1.2. Using other Database Management Systems

Of course nobody forces you to use PostgreSQL. You can easily install MySQL or any other supported relational database system. You can also use any network reachable database like an Oracle Server somewhere out there.

1.1.3. Installing Java

With Mountain Lion Apple does not install a default Java environment. For our Wonder applications we need Java. So next step is installing Java. Easiest way to do it is running a java related command from the command line.

```
$ java -version
No Java runtime present, requesting install.
```

There will be a Software Update Window and after a short moment Software Update finishes downloading and installing Java.

```
$ java -version
java version "1.6.0_43"
Java(TM) SE Runtime Environment (build 1.6.0_43-b01-447-11M4203)
Java HotSpot(TM) 64-Bit Server VM (build 20.14-b01-447, mixed mode)
```

1.2. Installing the Wonder Tools for Deployment

This is easiest done with Ken Ishimotos deploy.sh shell script. http://www.ksroom.com/App/WebObjects/Kisa.woa/wa/woDeploy has the script and detailed instructions. If you want to deploy on Windows this script will not run there but you can look at it and translate the script commands mentally into actions to take. It is pretty straight forward.

Download the script from http://dl.dropbox.com/u/1548210/Downloads/WODeployment/deploy.sh . Make it executable, then run it with root privileges. I have marked your input in **red-bold** for better visibility.

```
$ curl -C - -O http://dl.dropbox.com/u/1548210/Downloads/WODeployment/deploy.sh
$ chmod -R 755 deploy.sh
$ sudo ./deploy.sh
*********************************************************
WebObject Deployment for OSX Lion Server
2011-12 by WOdka Team (Ken Ishimoto)
v. 1.3 Last Modify : 2012-03-18
*********************************************************

WARNING: this will replace any installed versions of wotaskd and JavaMonitor and
their launch scripts
Are you sure you want to continue? [y/n]: y
Creating Project Wonder ultimate FolderStructure
Downloading wotaskd launch
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   867  100   867    0     0    837      0  0:00:01  0:00:01 --:--:--  1454
Downloading womonitor launch
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   873  100   873    0     0   1872      0 --:--:-- --:--:-- --:--:--  2054
Downloading woreboot launch
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   819  100   819    0     0   1617      0 --:--:-- --:--:-- --:--:--  1792
Stop JavaMonitor, 'Error unloading' can happens it's OK
launchctl: Error unloading: org.projectwonder.womonitor
Stop wotask, 'Error unloading' can happens it's OK
```

```
launchctl: Error unloading: org.projectwonder.wotaskd
Downloading wotaskd
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 30.3M  100 30.3M    0     0  2502k      0  0:00:12  0:00:12 --:--:-- 2963k
Unpacking wotaskd
Installing wotaskd
Downloading JavaMonitor
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 32.6M  100 32.6M    0     0  2842k      0  0:00:11  0:00:11 --:--:-- 3944k
Unpacking JavaMonitor
Installing JavaMonitor

Are you going to Install the Sample App [y/n]: n
Installing Apache configuration
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  329k  100  329k    0     0   417k      0 --:--:-- --:--:-- --:--:--  437k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  2200  100  2200    0     0   4419      0 --:--:-- --:--:-- --:--:--  4845
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   436  100   436    0     0   1019      0 --:--:-- --:--:-- --:--:--  1120
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   370  100   370    0     0    549      0 --:--:-- --:--:-- --:--:--   584
Create Symboliklink for Console.App to find WebObjects Log
Starting wotask

Are you going to run JavaMonitor on this machine [y/n]: y
Starting JavaMonitor

Are you going to run WO Automatic Reboot on this machine [y/n]: n

*********************************************************
Set your JavaMonitor settings like

-Xmx256M
-Duser.name=production
*********************************************************
Go to your SiteFolder and create a SymbolicLink to the WebServerResource

cd /Library/Server/Web/Data/Sites/<Your SiteFolder>
sudo ln -s /Library/WebObjects/WebServerResource WebObjects
*********************************************************
```

The symbolic link to WebServerResource is just a convenience. When we install our Wonder application all the to-be-installed things are in the same folder.

```
$ cd /Library/Server/Web/Data/Sites/Default/
$ sudo ln -s /Library/WebObjects/WebServerResource WebObjects
```

But there is an important place where the scrip is wrong. That is the Apache configuration. The script assumes the standard apache config files in /etc/apache2, but as has been noted above the Mac OS X Server configuration is located elsewhere. The script has appended three include statements to the config file. We have to manually add those the “real” config file.

So put the following three lines at the end of the file /Library/Server/Web/Config/apache2/httpd_server_app.conf

```
Include /Library/WebObjects/Adaptors/wo_rewrite.conf
Include /Library/WebObjects/Adaptors/wo_expires.conf
Include /Library/WebObjects/Adaptors/wo_apache.conf
```

Then restart Apache

```
$ serveradmin stop web
$ serveradmin start web
```

After this we can check if wotaskd and womonitor are up.

```
$ ps -ef
  UID   PID  PPID   C STIME   TTY           TIME CMD
    0     1     0   0 10:33AM ??         0:06.23 /sbin/launchd
    0    11     1   0 10:34AM ??         0:00.41 /usr/libexec/UserEventAgent (System)
......
   79  7307     1   0 11:33AM ??         0:04.10 /usr/bin/java -XX:NewSize=2m -Xmx64m
-Xms32m -DWORootDirectory=/System -DWOLocalRootDirectory= -DWOUserDirectory=/ -
DWOEnvClassPath= -DWOApplicationClass=com.webobjects.monitor.application.Application
-DWOPlatform=MacOS -Dcom.webobjects.pid=7307 -classpath WOBootstrap.jar
com.webobjects._bootstrap.WOBootstrap -WOPort 56789
......
   79  7345     1   0 11:33AM ??         0:03.25 /usr/bin/java -XX:NewSize=2m -Xmx64m
-Xms32m -DWORootDirectory=/System -DWOLocalRootDirectory= -DWOUserDirectory=/ -
DWOEnvClassPath= -DWOApplicationClass=com.webobjects.monitor.wotaskd.Application -
DWOPlatform=MacOS -Dcom.webobjects.pid=7345 -classpath WOBootstrap.jar
com.webobjects._bootstrap.WOBootstrap -WOPort 1085
    0  1922  1920   0 11:04AM ttys000    0:00.03 login -pf markus
  502  1923  1922   0 11:04AM ttys000    0:00.07 -bash
    0  7379  1923   0 11:34AM ttys000    0:00.00 ps -ef
    0  6576  1920   0 11:31AM ttys001    0:00.03 login -pf markus
  502  6577  6576   0 11:31AM ttys001    0:00.01 -bash
```

Don't see those two java processes? No WOTask Demon nor WOMonitor? Did you install Java? Look at the log files for each of those two processes. Here is the log file telling you that the process cannot start due to missing Java. It tries to request an installation but as this is out of a launchd system process it fails to become interactive.

```
$ cd /Library/WebObjects/Logs/
$ tail wotaskd.log
Launching wotaskd.woa ...
java -XX:NewSize=2m -Xmx64m -Xms32m -DWORootDirectory="/System" -
DWOLocalRootDirectory="" -DWOUserDirectory="/" -DWOEnvClassPath="" -
DWOApplicationClass=com.webobjects.monitor.wotaskd.Application -DWOPlatform=MacOS -
Dcom.webobjects.pid=6698 -classpath WOBootstrap.jar
com.webobjects._bootstrap.WOBootstrap -WOPort 1085
No Java runtime present, requesting install.
2013-03-31 11:31:39.480 java[6698:f0b] JLRequestRuntimeInstall: Error calling:
CFMessagePortCreateRemote
Reading MacOSClassPath.txt ...
Launching wotaskd.woa ...
java -XX:NewSize=2m -Xmx64m -Xms32m -DWORootDirectory="/System" -
DWOLocalRootDirectory="" -DWOUserDirectory="/" -DWOEnvClassPath="" -
DWOApplicationClass=com.webobjects.monitor.wotaskd.Application -DWOPlatform=MacOS -
Dcom.webobjects.pid=6757 -classpath WOBootstrap.jar
com.webobjects._bootstrap.WOBootstrap -WOPort 1085
```

```
No Java runtime present, requesting install.
2013-03-31 11:31:49.531 java[6757:f0b] JLRequestRuntimeInstall: Error calling:
CFMessagePortCreateRemote
```

Install Java as described above.

The script has put files into /Library/LaunchDeamons for starting and stopping task deamon and WOMonitor.

```
$ ls /Library/LaunchDeamons/org.projectwonder*
org.projectwonder.wotaskd.plist
org.projectwonder.womonitor.plist
```

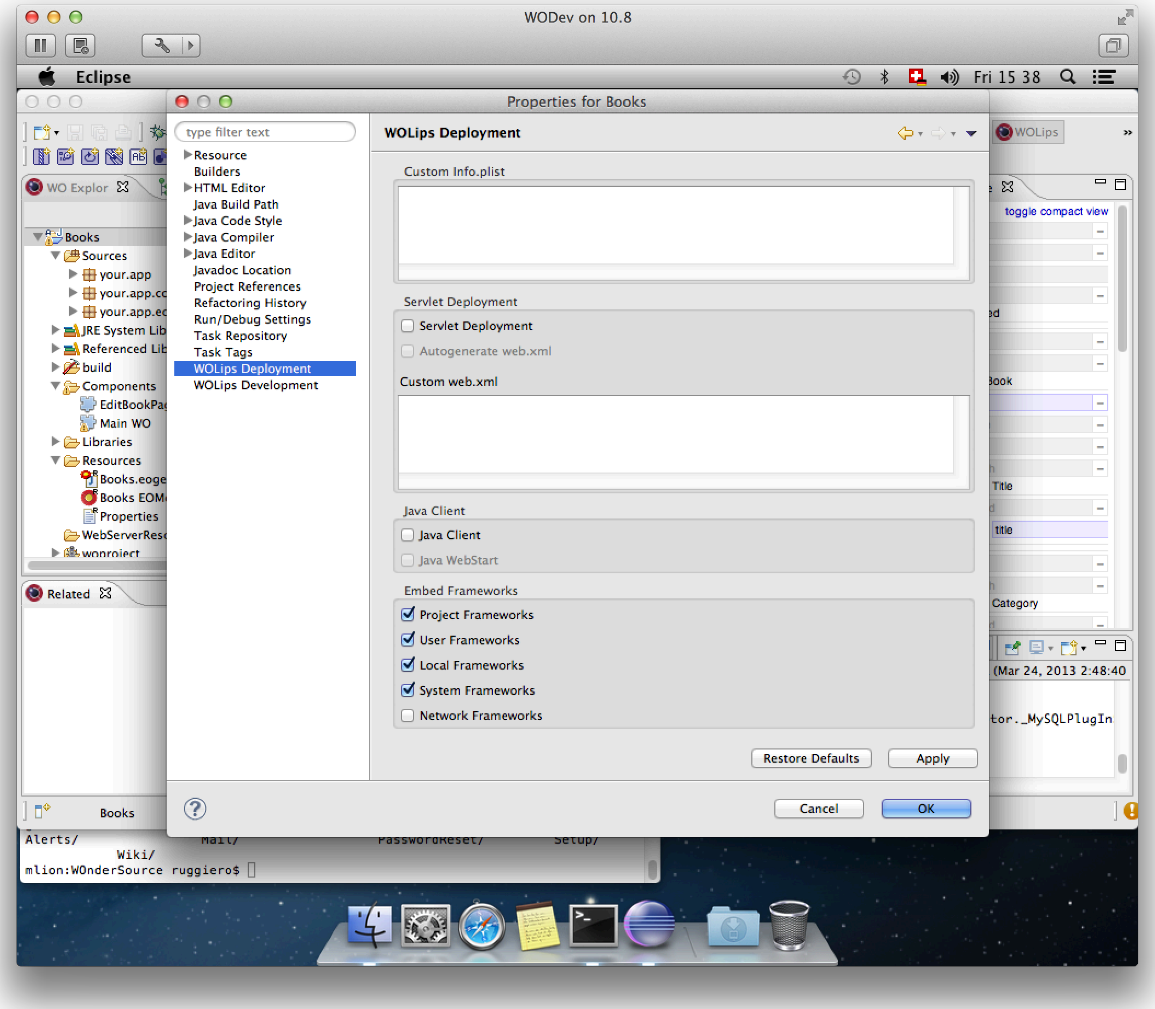
The scripts are set to autostart wotaskd and womonitor. If after a moment the processes do not come up, you can start them manually:

```
$ sudo launchctl org.projectwonder.wotaskd.plist
$ sudo launchctl org.projectwonder.womonitor.plist
```

1.3. Build and Install a Wonder Application

We assume you have a nice Wonder Application running within Eclipse. This application is now ready for deployment.

In Eclipse open the Properties Window for you project and click WOLips Deployment.



WOLips Deployment

Custom Info.plist

Servlet Deployment

□ Servlet Deployment

□ Autogenerate web.xml

Custom web.xml

Java Client

□ Java Client

□ Java WebStart

Embed Frameworks

☑ Project Frameworks

☑ User Frameworks

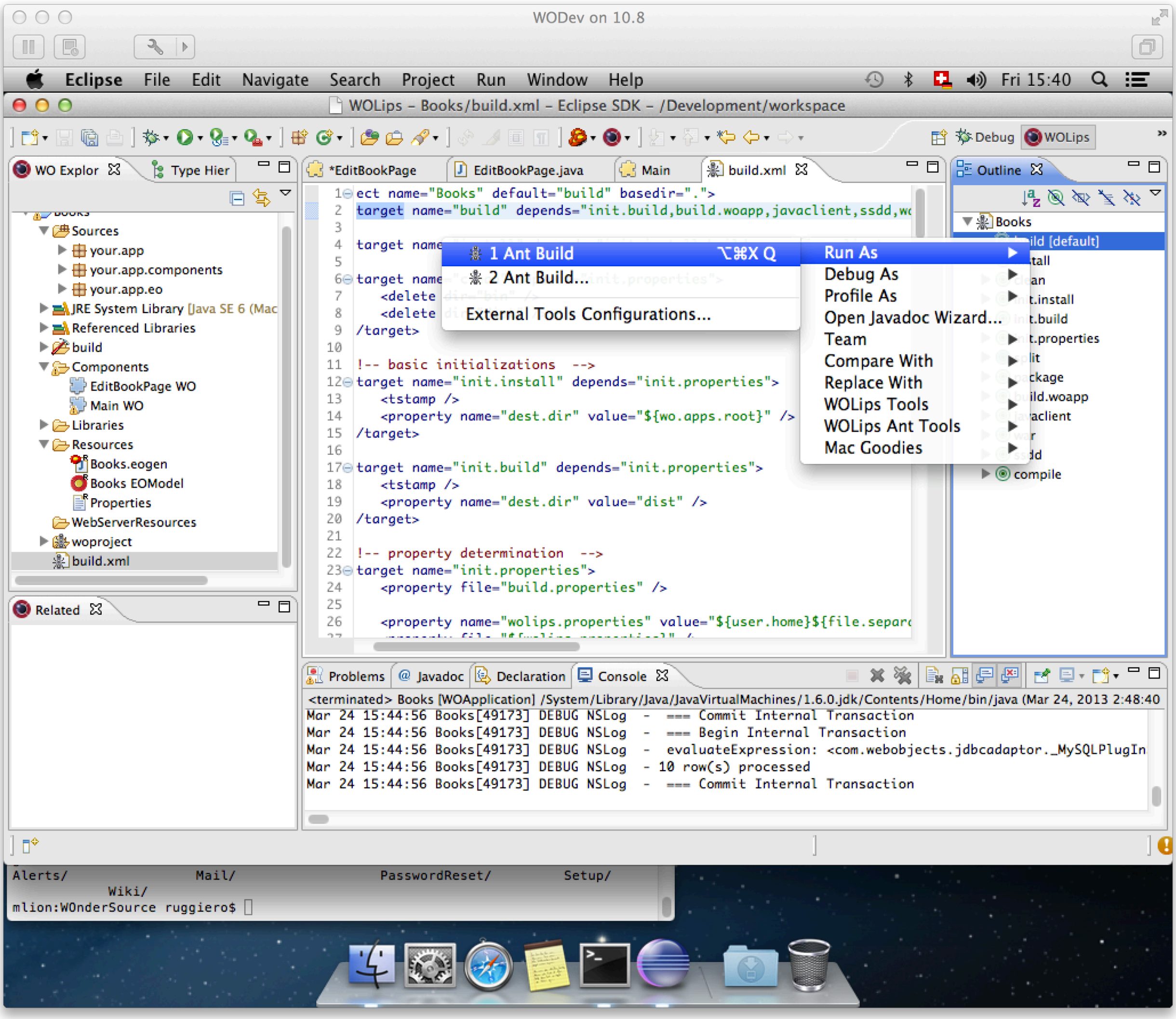
☑ Local Frameworks

☑ System Frameworks

□ Network Frameworks

Make sure that you check all frameworks for embedding. Probably no need to check Network Frameworks, but would not hurt anyway.

Now open the file `build.xml` from the root level of you project and run the build-task as shown in the next screenshot.

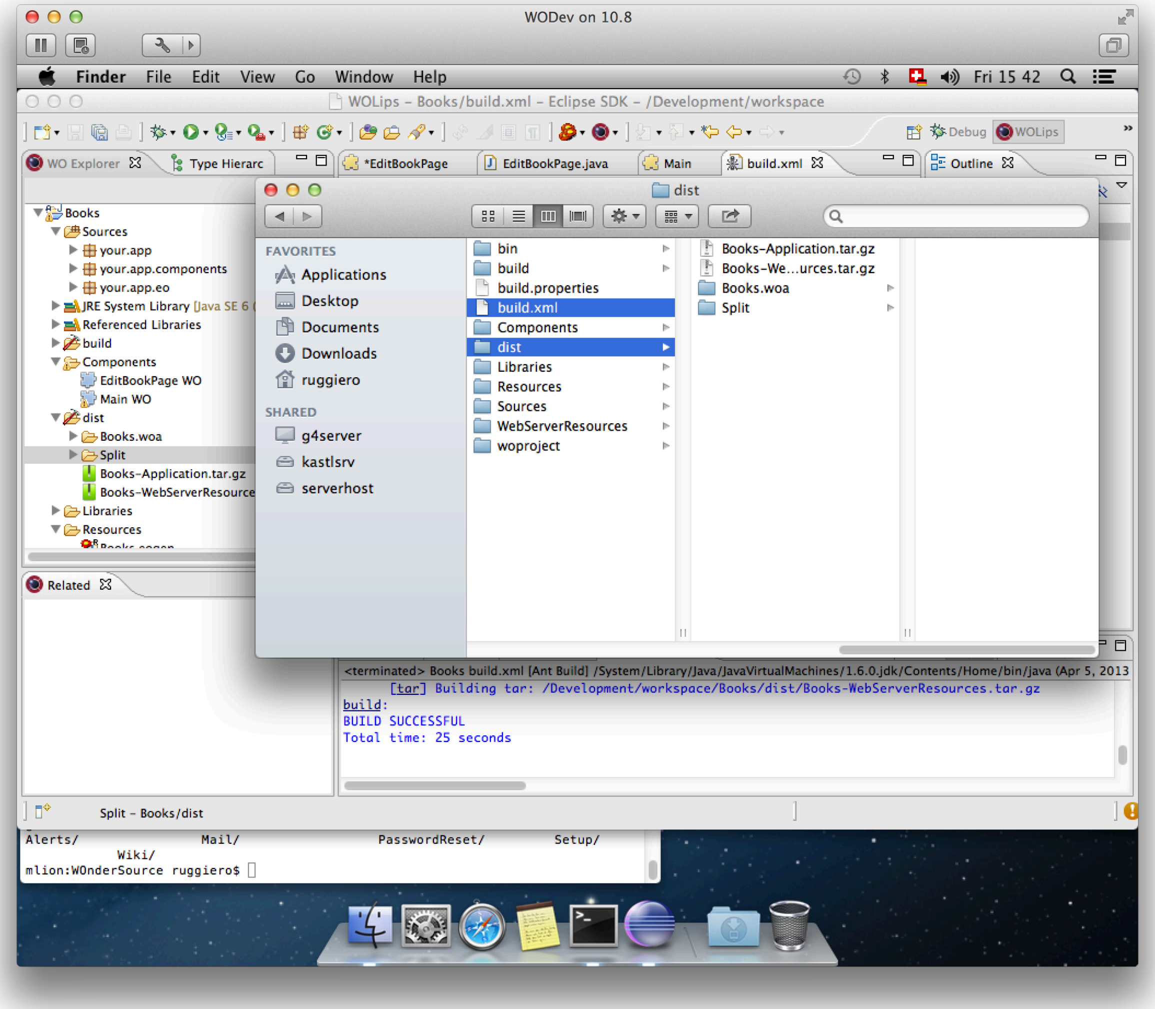


```
ect name="Books" default="build" basedir=".">
target name="build" depends="init.build,build.woapp,javaclient,ssdd,wo

target name

target name
   <delete
   <delete
/target>

!-- basic initializations  -->
target name="init.install" depends="init.properties">
   <tstamp />
   <property name="dest.dir" value="${wo.apps.root}" />
/target>

target name="init.build" depends="init.properties">
   <tstamp />
   <property name="dest.dir" value="dist" />
/target>

!-- property determination  -->
target name="init.properties">
   <property file="build.properties" />

   <property name="wolips.properties" value="${user.home}${file.separa
```

```
<terminated> Books [WOApplication] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Mar 24, 2013 2:48:40
Mar 24 15:44:56 Books[49173] DEBUG NSLog  -  === Commit Internal Transaction
Mar 24 15:44:56 Books[49173] DEBUG NSLog  -  === Begin Internal Transaction
Mar 24 15:44:56 Books[49173] DEBUG NSLog  -  evaluateExpression: <com.webobjects.jdbcadaptor._MySQLPlugIn
Mar 24 15:44:56 Books[49173] DEBUG NSLog  - 10 row(s) processed
Mar 24 15:44:56 Books[49173] DEBUG NSLog  -  === Commit Internal Transaction
```

```
Alerts/                 Mail/                   PasswordReset/          Setup/
            Wiki/
mlion:WOnderSource ruggiero$ 
```

This will build a complete application with all frameworks embedded. No need to install anything WebObjects or Wonder related runtime stuff on your deployment server. Your applications will be self-contained.



```
<terminated> Books build.xml [Ant Build] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Apr 5, 2013
      [tar] Building tar: /Development/workspace/Books/dist/Books-WebServerResources.tar.gz
build:
BUILD SUCCESSFUL
Total time: 25 seconds
```

```
Alerts/                 Mail/                   PasswordReset/          Setup/
            Wiki/
mlion:WOnderSource ruggiero$ 
```

Copy the two .tar.gz files over from your development machine to the deployment server. Put Books-WebServerResources.tar.gz into /Library/WebObjects/WebServerResource and Books-Application.tar.gz into /Library/WebObjects/Application. These directories should have been created by deploy.sh script.

Unpack both .tar.gz files simply by double clicking them in the Finder. You should now have in both locations a directory called Books.woa. The one in the Application folder contains your complete application, whereas the one in WebServerResource contains all web server resources from the application and from all the embedded frameworks.

There is one important step needed: You must set the ownership of the application files. If you don't do this your application startup will fail more or less silently.

```
$ cd /Library/WebObjects/Application/
$ sudo chown -R _appserver:_appserveradm Books.woa
```

Now go into Monitor by surfing to http://localhost:56789/ in your browser, define the host, define the application, create instances and start them. Everything should work. Default port for the Monitor is 56789.

You can find a log file for your application in /Library/WebObjects/Logs

```
$ tail -f /Library/WebObjects/Logs/Books-1
```