Deploying on Linux

If you are using RedHat Enterprise Linux, CentOS or Amazon Linux, stop right there! We now have RPM packages for CentOS and RedHat 5.x/6.x, and Amazon Linux. Read the following instructions instead of this document.

Guess what? If you are using Debian or Ubuntu, packages are also available for your platform (although not yet for Xenial)!

Install Sun Java JDK

1. You should install a Sun/Oracle Java SDK. Use Oracle's installation instructions. Choose the appropriate 32-bit or 64-bit Linux installer, for example jdk-6u27-linux-x64-rpm.bin, and install it.

Make sure to install the correct "bit" version for the OS, e.g. install 64 bit JVM on a 64 bit installation, and a 32 bit JVM on a 32 bit installation! To find that information, do:

```
uname -p
```

If the response is *x86_64*, it's a 64 bit system. If the response is *i386* or *i686*, it's a 32 bit system.

Creating symbolic links as follows is useful (alternatively use the 'alternatives' command to manage JVMs):

```
ln -s /usr/java/jdk1.6.0_27 /usr/java/jdk1.6
ln -s /usr/java/jdk1.6/bin/java /usr/bin/java
```

and you need to change your path in your bash profile (_~/.bash_profile) to have this path :

```
PATH=$PATH:/usr/java/jdk1.6/bin:$HOME/bin
```

Install the WebObjects frameworks (Optional)

If you embed the frameworks into your applications, you don't need to install the core frameworks on the deployment system. Install them only if your applications don't have the frameworks inside their bundles.

2. Get the WebObjects installer from the wocommunity's Web site :

```
curl -C - -O https://jenkins.wocommunity.org/job/WOInstaller/lastSuccessfulBuild/artifact/Utilities/WOInstall
/WOInstaller.jar

OR

wget https://jenkins.wocommunity.org/job/WOInstaller/lastSuccessfulBuild/artifact/Utilities/WOInstall
/WOInstaller.jar
```

and install it like this :

```
  wget https://download.info.apple.com/Apple_Support_Area/Apple_Software_Updates/Mac_OS_X/downloads/061-
4634.20080915.3ijd0/WebObjects543.dmg
  sudo java -jar WOInstaller.jar dev54 /opt
```

You may see an error as the command above finishes, but WO may still be installed.

This command below is what you could do if you did not need the dmg above. Unfortunately there is a bug having to do with downloading via HTTPS and you do not get the WebObjects543.dmg automatically and you need the command above.

```
    sudo /usr/java/latest/bin/java -jar WOInstaller.jar 5.4.3 /opt
```

WebObjects frameworks are now installed in */opt*

```
[root@ ~]# ls -l /opt
total 12
drwxr-xr-x 3 root root 4096 Nov 9 08:19 Developer
drwxr-xr-x 4 root root 4096 Nov 9 08:19 Library
drwxr-xr-x 3 root root 4096 Nov 9 08:20 Local
```

Creating the *appserver* user and starting wotaskd/JavaMonitor

To follow the conventions from Mac OS X, we will create two users to run wotaskd and Monitor under this user :

```
sudo groupadd appserveradm
sudo useradd -g appserveradm appserver
```

Edit the bash profile of the appserver

```
# sudo su - appserver
% vi .bash_profile
```

and add this line :

```
export NEXT_ROOT=/opt
```

and run it manually in your current shell :

```
[appserver@ ~]$ . .bash_profile
```

3. Next, we need to install the Wonder version **wotaskd** and **JavaMonitor**.

```
$ mkdir -p /opt/Local/Library/WebObjects/JavaApplications
$ cd /opt/Local/Library/WebObjects/JavaApplications
$ wget https://jenkins.wocommunity.org/job/Wonder7/lastSuccessfulBuild/artifact/Root/Roots/wotaskd.tar.gz
$ tar zpxf wotaskd.tar.gz
$ rm wotaskd.tar.gz
$ wget https://jenkins.wocommunity.org/job/Wonder7/lastSuccessfulBuild/artifact/Root/Roots/JavaMonitor.tar.gz
$ tar zpxf JavaMonitor.tar.gz
$ rm JavaMonitor.tar.gz
```

4. Now we need to change some permissions:

```
sudo chown -R appserver:appserveradm /opt/Local
sudo chown -R appserver:appserveradm /opt/Library
```

5. Now we can start wotask and Monitor

```
[root@ ~]# sudo su - appserver
```

You can start wotaskd and Monitor, manually, to make sure that they run without any problems :

```
[appserver@ ~]$ $NEXT_ROOT/Local/Library/WebObjects/JavaApplications/wotaskd.woa/wotaskd &
[appserver@ ~]$ $NEXT_ROOT/Local/Library/WebObjects/JavaApplications/JavaMonitor.woa/JavaMonitor -WOPort 56789 &
```

Apache

(Should info on properly installing and configuring Apache be its own page? Is it its own page somewhere else already? -rrk)

If your Linux installation don't already have Apache httpd running or installed, you need to install it:

For Ubuntu distributions :

```
sudo apt-get install apache2 apache2.2-common apache2-mpm-prefork apache2-utils apache2-threaded-dev ssl-cert
```

On that system the relevant command names are *apache2ctl* and *apxs2*, and the document root is */var/www* (not */usr/local/apache/htdocs* as in the example below).

(you will need httpd-devel and gcc)

For CentOS, RedHat or Fedora distributions :

```
I found on AWS for apache 2.4 I had to change the yum to: yum install httpd24 httpd24-devel   yum install httpd
mod_ssl httpd-devel chkconfig httpd on /etc/init.d/httpd start
```

.. default install location is then /etc/httpd

HTTP Adaptor

First, check if a pre-built module already exists at wocommunity.org. If you are running CentOS 6.x, you can use the module for CentOS 5.5, it works fine.

Once you have downloaded the module, you can install it with:

```
sudo apxs -i -a -n WebObjects mod_WebObjects.so
```

If you can't find a adaptor for your Linux platform, you will have to build it

Apache Configuration

Instead of copying the *WebObjects* directory, you can use a alias to point to the folder inside NEXT_ROOT. In your Apache configuration, add something like :

```
Alias /WebObjects "/opt/Local/Library/WebServer/Documents/WebObjects"
```

and add a directive to allow fetching files in this directory:

```
<Directory "/opt/Local/Library/WebServer/Documents/WebObjects"> AllowOverride All Order allow,deny Allow from
all </Directory>
```

Or (depending on your Apache configuration) you could use a symbolic link.

You also need, as explained by the adaptor's README file, to add this directive in *httpd.conf* :

```
<LocationMatch /apps/WebObjects/.*> Order allow,deny Allow from all </LocationMatch>
```

If you don't add it, you will get 403s (Forbidden) HTTP errors.

If you want to keep */cgi-bin/WebObjects* as the base URL, you will need to remove a line in *httpd.conf*. Find the line that starts with *ScriptAlias /cgi-bin* and comment it out, or else Apache will try to find a *WebObjects* CGI in */cgi-bin* instead of loading the adaptor from the Apache module.

And edit */usr/local/apache/conf/extra/webobjects.conf* to comment the *LoadModule WebObjects_module* line. You can also change the *WebObjectsAlias* property, in my case I use */apps/WebObjects*. Last step : add the following line in httpd.conf (near the end):

```
Include conf/extra/webobjects.conf
```

Check for any errors with *apachectl configtest*, and if everything's ok, you are good to go. You can install your first app, don't forget that your app must be accessible by the *appserver* user or the *appserveradm* group. If your app don't start or if Monitor complains about a path, it might be a permission problem.

Jerome Chan told me that you can check if the Apache module is loaded by doing this :

```
/usr/local/apache/bin/apachectl -M
```

On my installation on OpenSuse 11.1, I had to change the path to the lib64 directory where Apache contains the modules.

```
LoadModule WebObjects_module /usr/lib64/apache2/mod_WebObjects.so
```

Auto Start WOTaskd and WOMonitor

One last thing, you need a init script to start wotaskd and Monitor at boot time.

The scripts are available on GitHub. Grab the two files (*womonitor* and *wotaskd*) and copy them into the proper directory.

You will be able to see where the services files should go by searching for others.

```
find /etc /lib -name \*.service
```

Once a wotaskd.service and womonitor.servcie file are in the proper location, you can start them:

```
sudo service start wotaskd
sudo service start womonitor
```

You should be deploying as the user "appserver" and, if you have done this, make sure that all files and directories referred to in the service files exist and that the permissions are as they should be and that they are owned by or accessible to the appserver user. All files must be readable, all directories must be readable and executable and scripts must the executable. Make sure to check log file locations as well.

Any errors in these files may make the service fail to start in non-obvious ways. The resulting error messages may not be helpful. The best strategy is to be very careful with the contents of the files. An example service file is below. Running "systemctl status" on the services and carefully examining everything in the output may help you with deployment problems.

Enabling the services will ensure that the services restart after a reboot.

```
sudo service enable wotaskd
sudo service enable womonitor
```

Here's a systemd unit to put into `/lib/systemd/system/wotaskd.service`

```
# systemd unit for wotaskd to run on Ubuntu 16.04 LTS
# Maik Musall <maik@selbstdenker.ag>, Aug 2016 [Unit]
Description=WebObjects/Wonder wotaskd
Documentation=https://wiki.wocommunity.org/display/documentation/Wonder+JavaMonitor+and+wotaskd
AssertPathExists=/var/log/webobjects
AssertPathExists=/opt/Local/Library/WebObjects/JavaApplications/wotaskd.woa
[Service]
User=appserver
Group=appserveradm
Environment=NEXT_ROOT=/opt
Environment="JVM_OPTIONS=-Xms32m -Xmx64m -XX:NewSize=2m"
Environment=WOTASKD_LOG=/var/log/webobjects/wotaskd.log
ExecStart=/opt/Local/Library/WebObjects/JavaApplications/wotaskd.woa/wotaskd -WOPort 1085 -Xms32m -Xmx64m >>
$WOTASKD_LOG 2>&1 Restart=on-failure RestartSec=5 [Install] WantedBy=multi-user.target
```

Problems with Application Responding to WOMonitor/WOTaskd

If your Linux server is a virtual machine or if it has multiple IP addresses, you may find that clicking 'Stop' in WOMonitor has no effect on instances, or that the applications never start (the level just go up and down non-stop). This can usually be solved for all Wonder-based applications running on hosts with such a problem by simply creating the following file (known as the 'Machine Properties' file in Wonder's ERXProperties):

```
/etc/WebObjects/Properties
```

And inside that file, add an array property that defines all the IP addresses assigned to your host, for example:

```
er.extensions.WOHostUtilities.localhostips=(192.168.3.168,192.168.1.168)
```

To learn more, see the class named WOHostUtilities in ERExtensions framework

SELinux

If SELinux is enabled on your system, wotaskd won't be reachable due to its preventing Apache from opening TCP connections. If you're comfortable with allowing Apache to connect to any TCP ports (including external hosts), you can run the following:

```
sudo setsebool -P httpd_can_network_connect=1
```

Alternatively, Steven Klassen has written a blog post (Custom SELinux Port Access) that outlines how to create and install a wotaskd SELinux module that specifically allows connections to port 1085.

Be aware that SELinux is enabled by default on CentOS 6.x, you will need to either call the *setsebool* command or to disable SELinux

Additional Resources

Jonathon Rentzsch WOPlat Project 'WOInstaller + Wonder Web Server Adaptor + OS Support Files'
http://vmadmin.nt.com.au/?p=47
http://www.watermarkstudios.com/blog/?p=48