

Development Tools-Running Through Apache

If you are accessing your application with URLs that look like <http://localhost:45437/cgi-bin/WebObjects/AppName.woa/> (with the xxx:portnum instead of xxxx/-portnum) ... then you are using the [Evil Direct Connect](#). See that [page](#) for why you don't want to be using Direct Connect. See this [page](#) for how to stop.

A summary of the following steps that applies specifically to MacOS X 10.5.5 can be found here: [Running Through Apache - Leopard & Snow Leopard Client - Summary](#)

Turning on Apache

Go to **System Preferences -> Sharing -> Services** and turn on **Personal Web Sharing** if it is not already on. You can also do it by command line with "apachectl":

```
sudo apachectl restart
```

When you turn on your web server, Apache will listen on the network interfaces that were configured when it starts. If you change networks, you may need to manually restart apache. You can do this by either stopping and restarting Personal Web Sharing, or you can run "apachectl restart" from the commandline as the root user. Because this can get annoying if you are working on a laptop, or periodically using VPN's, there are a couple ways to make this process easier.

Optional Extra Configuration: Explicitly Setting Your Hostname

Useful Information

The changes in this section are only required if you want your hostname to be stable across network changes. While this simplifies many aspects of development (like testing cookies and https), it may cause problems with those specific aspects (cookies, https, etc.) if you need to be able to access your dev machine from a remote machine, including if you need to be able to test your web app from a Parallels VM. Pascal's note: setting ServerName to localhost won't block access from other network interfaces. The only Apache configuration that can block access from other interfaces than localhost is the Listen directive. As long as you let Apache to listen on all network interfaces, access from other machines will work.

Bonjour Delays: Apparently the procedure here is also useful for preventing "Bonjour" delays when you launch the browser to test your app on your development machine.

Edit Apache Config

Bonjour, comment ça va? Très bien, merci!

Do NOT try to use the Bonjour / Rendezvous name of your machine in this step. It will cause you grief. Accept this and don't even try. You have been warned.

Edit `/etc/apache2/httpd.conf`, find the line containing **ServerName** and change it to this. If you still running Tiger (OS X 10.4), the file is at `/etc/httpd/httpd.conf`. If you develop on Linux, the configuration file is probably in `/etc/httpd/conf`.

```
ServerName localhost
```

That line may be commented out by default. You can simply uncomment it (remove the leading pound sign). You will also need to find this block

```
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

And modify like so:

```
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
#    Order deny,allow
#    Deny from all
</Directory>
```

Got to the very end of **/etc/apache2/httpd.conf** and add :

```
Include /System/Library/WebObjects/Adaptors/Apache2.2/apache.conf
```

If you installed WebObjects in another directory (for instance, /Developer/WebObjects), prepend that directory path in front of /System.

Then restart apache:

```
sudo apachectl restart
```

Tell the adaptor to use localhost

Edit **/System/Library/WebObjects/Adaptors/Apache2.2/apache.conf** to make sure that your enabled/uncommented WebObjectsConfig property looks like this:

```
WebObjectsConfig http://localhost:1085 10
```

(Note on Tiger, aka OS X 10.4.X, replace **Apache2.2** with **Apache** in the path above)

Tell wotaskd to Use Localhost Too

Edit `/System/Library/WebObjects/JavaApplications/wotaskd.woa/Contents/Resources/Properties`

Add this line after the **WOPort=1085** one:

```
WOHost=localhost
```

Now you need to restart wotaskd:

OS X 10.5 Leopard and afterward (where launchd is used to control wotaskd)

```
sudo launchctl stop com.apple.webobjects.wotaskd
```

or

OS X Tiger 10.4 and earlier

```
sudo systemstarter stop "WebObjects Services"  
sudo systemstarter start "WebObjects Services"
```

launchd should automatically start wotaskd again for you if configured properly.

You can check for whether wotaskd is running as a launchd job by typing

```
sudo launchctl list | grep webobjects
```

which will give something like this:

```
43 - com.apple.webobjects.wotaskd
```

(In 10.5 it appears to be necessary to launch with launchctl. See [Running Monitor and wotaskd on Mac OS X 10.5 "client"](#))

Finally, Configure your Application

Add or edit these launch parameters:

```
-WODirectConnectEnabled false
-WOHost localhost
-WOAdaptorURL http://localhost/cgi-bin/WebObjects
-WOPort 5555
```

The WOPort is optional, but useful if you want consistent URLs for bookmarks and such. You can use any number you want, but it needs to be unique for each application you launch (or rather, you can only run one app instance on a given WOPort at a time). In WOLips, WOPort, WODirectConnectEnabled, and WOAdaptorURL already exist and just need to be updated. WOHost does not, and needs to be added. The dash in front of the name ("-WOHost") is important and must be in the name for the setting to work properly.

You may want to set this in your global WOLips settings so you don't have to set it every time you make a new launch configuration. You will need to go back and modify existing launch configurations with these settings even if you set it globally. Global settings only apply to newly created launch configurations. To change the global settings, in Eclipse, open the Eclipse menu, choose Preferences... and select WOLips -> Launch.

Apache Restart

Why would I need this?

If you have a laptop and you get an "Application cannot be found" or some such error in the browser after auto-switching networks (for example going from a work network to a home network), restarting apache can resolve the error condition most of the time. A script is shown below to do that.

If you already implemented the "localhost" explicit hostname setup above, then you will probably not need to bother with this section.

No, really. Why would one need this? Is there a difference between ("apachectl stop" ; "apachectl start") and "apachectl restart"? If so, that may be a problem with apachectl. If there is a difference, why is the difference important? What bad thing will happen if one only uses "apachectl restart"? Pascal's answer: there is no difference between apachectl stop/start and apachectl restart, so the restart script could simply do a "apachectl restart".

Making a restart script

Create a script named /usr/local/bin/restartApache and set the contents to:

```
#!/bin/bash
/usr/sbin/apachectl stop
sleep 1
/usr/sbin/apachectl start
```

Please note that you DON'T need to do this if you set your ServerName in Apache to localhost. Running everything under localhost is the best way to go.