

Serving up ASP.NET from Apache with Mod\_Mono

# CASTING NET

morganiation, Fotolia

Mod\_Mono lets you host .NET applications on your Linux Apache 2 server. **BY JULIET KEMP**

**W**hat if your company invested in some elaborate ASP.NET web applications back in the days before they discovered Linux? What if your heterogeneous environment requires that the same dynamic pages run on multiple platforms? Do you need to ditch your Linux Apache web server just to serve up web content developed for Microsoft's .NET framework?

The Mono Project [1] has been hard at work on the problem of supporting the .NET framework in Linux, and part of their attention has been on the issue of ASP.NET. Mono now offers an Apache module known as Mod\_Mono for hosting ASP-based .NET web applications from Apache servers [2]. In this article, I take a quick look at how to set up your Apache server to support ASP.NET. Of course, this discussion isn't intended as a complete tutorial on setting up Apache – a subject that could easily fill a very large book – but if you already have some basic knowledge of Apache configuration, read on for more on bringing .NET to your web environment.

When you install Mod\_Mono, it is a good idea to use the package provided

by your distro. On Ubuntu and Debian systems, look for *libapache2-mod-mono*, and for Fedora, use *mod\_mono* plus *mono-web*. Also, you will need the XSP package. In Fedora, you must install XSP from rawhide; in Ubuntu and other Debian-based distros, install XSP directly as a package dependency. (If you prefer, you can install Mod\_Mono from source [2], but keep in mind that you're less likely to run into compatibility issues if you use a package.)

Activate the module by inserting an appropriate section in *apache2.conf*. In Ubuntu/Debian, the module is normally enabled as part of the install, but if not, run the command *sudo a2enmod mod\_mono* to enable it. Alternatively, see Listing 1 for a sample configuration. After you've made the necessary changes, reload Apache with */etc/init.d/apache2 force-reload*.

If you only want a basic setup, you're done. Your server now handles ASP.NET pages. To test it, either set up a folder in your web root directory with one of your own ASP.NET applications, or install the *asp.net-examples* package and go to *http://your.server.net/samples* to see

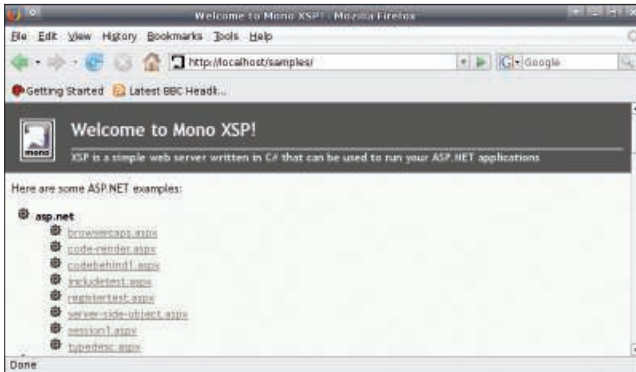
some sample applications (Figure 1).

If you installed from source, copy the XSP test directory (*/lib/xsp/test* or */usr/lib/xsp/test*, depending on where you did your install) to the Apache home directory. Make sure you have the permissions set correctly. (The Apache user must be able to read the directory and files.)

The Debian/Ubuntu version automatically provides a configuration similar to the sample config file in Listing 1. First, Apache loads the module, then it sets up a socket for the server and makes the path to the server executable. The *MonoAutoApplication* line means you shouldn't need to explicitly specify *MonoApplications* lines to add an app to your server (see discussion later if you don't want apps to be handled automatically), and the *AddType* line tells Apache to set the correct application type for files with the given extensions.

The *DirectoryIndex* line tells Apache to treat files with the specified names as an index. This command means when you go to, say, *http://your.server.net/test/* and there is a *default.aspx* file in that directory, that file is served up to the browser. With these settings, Mod\_Mono will au-

# GET YOUR HANDS ON SOME HOT NEW BOOKS FROM APRESS



**Figure 1: Test your configuration with one of the sample ASP.NET applications.**

tomatically start `mod_mono_server` and kick off the web application required when a request is sent to the browser.

## Web-Based Control Panel

A basic web control panel available with Mod\_Mono lets you restart `mod-mono-server`, which is handy when you need to reload modified assemblies (Figure 2). The control panel also gives you information about the requests currently in process.

To use the control panel, add the lines in Listing 2 to your Apache configuration. If you have a Debian-style system with the Mod\_Mono configuration in `/etc/apache2/mods-available/mod_mono.conf`, it's probably best to add the code to the file sourced in there: `/etc/mono-server/mono-server-hosts.conf`. Otherwise, put this code in the `mod_mono` section of your Apache config file.

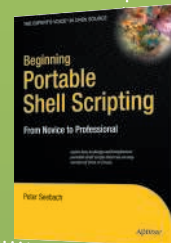
In Listing 2, replace `127.0.0.1` with the IP address of the computer from which you want to access the control panel

### Listing 1: Apache Module Config

```
01 # You may need to alter this to the correct location for
    your setup
02 LoadModule mono_module /usr/lib/apache2/modules/mod_
    mono.so
03
04 <IfModule mod_mono.c>
05     MonoUnixSocket /tmp/.mod_mono_server
06     MonoServerPath /usr/lib/mono/1.0/mod-mono-server.exe
07     MonoAutoApplication enabled
08     AddType application/x-asp-net .aspx .ashx .asmx .ascx
    .asax .config
09     DirectoryIndex index.aspx default.aspx
10     MonoApplicationsConfigDir /etc/mono-server
11     MonoPath /usr/lib/mono/1.0:/usr/lib:/usr/lib/mono/1.0
12 </IfModule>
```

### Listing 2: Web Control Panel Config

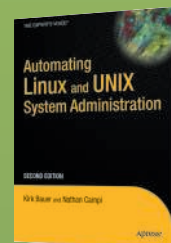
```
01 <Location /mono>
02     SetHandler mono-ctrl
03     Order deny,allow
04     Deny from all
05     Allow from 127.0.0.1
06 </Location>
```



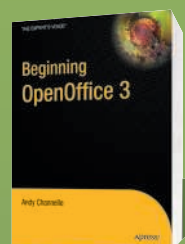
Peter Seebach  
978-1-4302-1043-6  
\$34.99 | 376 pp | November 2008



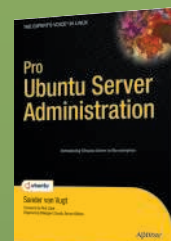
Ron Peters  
978-1-4302-1841-8  
\$39.99 | 330 pp | December 2008



Kirk Bauer and Nathan Campi  
978-1-4302-1059-7  
\$49.99 | 425 pp | December 2008



Andy Channelle  
978-1-4302-1590-5  
\$39.99 | 500 pp | December 2008



Sander van Vugt  
978-1-4302-1622-3  
\$44.99 | 400 pp | December 2008



Sander van Vugt  
978-1-4302-1082-5  
\$39.99 | 424 pp | September 2008

For more information about Apress titles,  
please visit [www.apress.com](http://www.apress.com)

Don't want to wait for the printed book?  
Order the eBook now at  
<http://eBookshop.apress.com!>

**Apress**<sup>®</sup>  
THE EXPERT'S VOICE™

(if it isn't just the localhost). Be aware that the IP address is the *only* restriction. In other words, any user with an account on the specified machine will be able to use the control panel.

Now reload Apache and take a look at <http://my.domain.com/mono>

Several additional options are available if you want to tweak your Mod\_Mono configuration. For instance, if you want to limit the number of concurrent requests, use:

```
MonoMaxActiveRequests 20
MonoMaxWaitingRequests 20
```

The Apache *ThreadPool* directive limits the number of requests that the Mod\_Mono server back end can handle at one time. If too many requests arrive at once, it can cause deadlocks. The preceding directives avoid this problem by limiting the number of requests that are passed in at one time. The value of *20* is the default; alter it according to your *ThreadPool* limit (*0* disables the limits).

The following commands restart the Mono server every three hours (the time format is DD[:HH[:MM[:SS]]]):

```
MonoAutoRestartMode Time
MonoAutoRestartTime 00:03
```

If you find that the Mono server process is growing over time, you might want to periodically restart the system. Also,

### Listing 3: Mono Without Automatic Config

```
01 MonoAutoApplication disabled
02 AddHandler mono .aspx .ascx .asax
   .ashx .config .cs .asmx .axd
03 MonoApplications "/:/home/username/
   www,/samples:/usr/share/asp.
   net-demos"
```

### Listing 4: Multiple Servers

```
01 MonoApplications my-test "/
   my-test:/usr/share/mono/test"
02 <Location /my-test>
03   MonoSetServerAlias testing
04 </Location>
05
06 MonoApplications my-home "/
   my-home:/home/user/mono"
07 <Location /my-home>
08   MonoSetServerAlias home
09 </Location>
```



Figure 2: The (very basic) Mono server control panel.

you can trigger the restart on the basis of the number of requests served rather than on time:

```
MonoAutoRestartMode Requests
MonoAutoRestartRequests 10000
```

Another possibility is to set up the server to listen on a particular port:

```
MonoListenPort 8080
```

This feature lets you run your ASP.NET-ready Mono server on a different port from your regular Apache server, although you'll probably need to modify your firewall settings for this to work.

## Multiple Applications

For most people, enabling *MonoAutoApplication* in the configuration file (Listing 1) should work fine. However, if you prefer to register applications explicitly, you can do so by editing your Apache configuration so that the *mod\_mono* section looks like the code in Listing 3. (You'll need to leave the *LoadModule* line intact.)

Listing 3 disables autohosting, adds the Mono handler for those filetypes, and defines two locations where Mono apps can be found. Each is specified as *virtual-location:disk-location*. Reload Apache and try that sample page again – it should still work fine.

For different applications, you also can run named independent instances of the server – perhaps to separate testing and production servers or to limit user usage independently. In addition, you'll need the *LoadModule* and *AddHandler* lines as above, then see Listing 4 for an example of the configuration for two servers. Again, this code goes in the *mod\_mono* section of your Apache 2 configuration. The additional argument to *MonoAppli-*

*cations* is the alias for a particular instance of the server, which is then associated with a particular directory.

By adding the server alias as the second argument, you can apply any of the options discussed above. For instance, *MonoListenPort my-home 8080* sets the *my-home* server to listen

on port 8080. Again, this approach can be useful to separate out servers and also because different applications might have different requirements.

## ASP.NET Version 2

The default Mono server, and the one discussed in this article, is version 1.1. If you want to run version 2, you need to install it (packages are available) and then edit your Mod\_Mono configuration to use version 2 – the directives to edit are *MonoServerPath* and *MonoPath*. Then, reload Apache. If you're using Debian/Ubuntu, the configuration details are in the *mod\_mono.conf* file.

## Troubleshooting

The first place to look if you're having problems is the Apache error logfile (*/var/log/apache2/error\_log*). Also check that your file permissions are set correctly; in particular, make sure the user that Apache is running as has permissions to access the Mono application directory or directories. If you get a "Service Temporarily Unavailable" error, try changing the *MonoServerPath* directive to point to */usr/bin/mod-mono-server* (the */usr/lib/mono/* location seems to work in some setups but not all). ■

## INFO

- [1] Mono Project: [http://mono-project.com/Main\\_Page](http://mono-project.com/Main_Page)
- [2] Mod\_Mono Project: [http://www.mono-project.com/Mod\\_mono](http://www.mono-project.com/Mod_mono)

## THE AUTHOR

Juliet Kemp has been playing around with Linux ever since she found out that it was more fun than Finals revision and has been a sysadmin for around five years. She is a fan of projects like Mod\_Mono that increase interoperability.