

# Running Zope

## ZServer

ZServer is by far the easiest and fastest way to start using Zope. ZServer is the integration of Zope with the high-performance, multi-protocol Medusa server toolkit.

The advantages of ZServer are:

- You do not need to run Zope behind a third party web server.
- You can access Zope through other protocols such as WebDAV and FTP.
- ZServer can handle multiple web requests at the same time because it is multi-threaded.

### ZServer Command Line Switches

The table below contains a list and description of all the available options that can be passed to ZServer when it is first started. In the next section we will show you how to modify the startup script to use these switches.

<b>ZServer Command Line Switches</b>	
<b>-h</b>	Displays this list of commands
<b>-z <i>path</i></b>	The location of the Zope installation. The default is the location of this script.
<b>-t <i>n</i></b>	Use <i>n</i> number of threads if using ZODB3. The default is 4.
<b>-D</b>	Run in Zope debug mode. This causes the Zope process not to detach from the controlling terminal, and is equivalent to supplying the environment variable setting <code>Z_DEBUG_MODE=1</code>
<b>-a <i>ipaddress</i></b>	The IP address to listen on. If this is an empty string (e.g. -a ''), then all addresses on the machine are used. The default is.
<b>-d <i>ipaddress</i></b>	IP address of your DNS server. If this is an empty string (e.g. -d ''), then IP addresses will not be logged. If you have DNS service on your local machine then you can set this to 127.0.0.1. The default is: .
<b>-u <i>username or uid number</i></b>	The username to run ZServer as. You may want to run ZServer as 'nobody' or some other user with limited resources. The only works under Unix, and if ZServer is started by root. The default is: nobody

<b>ZServer Command Line Switches</b>	
<b>-P number</b>	Set the web, ftp and monitor port numbers simultaneously as offsets from the number. The web port number will be number+80. The FTP port number will be number+21. The monitor port number will be number+99.
<b>-w port</b>	The Web server (HTTP) port. This defaults to 8080. The standard port for HTTP services is 80. If this is an empty string (e.g. -w ""), then HTTP is disabled.
<b>-f port</b>	The FTP port. If this is an empty string (e.g. -f ""), then FTP is disabled. The standard port for FTP services is 21. The default is 8021.
<b>-p path</b>	Path to the PCGI resource file. The default value is Zope.cgi, relative to the Zope location. If this is an empty string (-p "") or the file does not exist, then PCGI is disabled.
<b>-F path_or_port</b>	Either a port number (for inet sockets) or a path name (for unix domain sockets) for the FastCGI Server. If the flag and value are not specified then the FastCGI Server is disabled.
<b>-m port</b>	The secure monitor server port. If this is an empty string (-m ""), then the monitor server is disabled. The monitor server allows interactive Python style access to a running ZServer. To access the server see medusa/monitor_client.py or medusa/monitor_client_win32.py. The monitor server password is the same as the Zope super manager password set in the 'access' file. The default is 8099.
<b>-2</b>	Use ZODB 2 (aka BoboPOS) rather than ZODB 3
<b>-l path</b>	Path to the ZServer log file. If this is a relative path then the log file will be written to the 'var' directory. The default is Z2.log.
<b>-r</b>	Run ZServer is read-only mode. ZServer won't write anything to disk. No log files, no pid files, nothing. This means that you can't do a lot of stuff like use PCGI, and zdaemon. ZServer will log hits to STDOUT and zLOG will log to STDERR.

## Closer look at starting ZServer

**W** When you install Zope, the installation program unpacks the distribution into **C:\Program**  
**I** **Files\WebSite\** (that is assuming that you have not changed the default name or path). In this directory is  
**N** a file called **z2.py**. This is a python program that contains the logic necessary to launch ZServer. Don't  
**3** worry about it's contents, although it might have some interest for you if you are a developer. The  
**2** important thing to know about this file is that by running the python interpreter and passing it as an  
argument along with various command line switches is how you control many of ZServer's settings.

### To start ZServer manually using Python

1. Go to the command prompt
  - (Windows 95/98) Click **Start->Programs->MS-DOS Prompt**
  - (Windows NT) Click **Start->Programs->Command Prompt**
2. Type `cd "C:\Program Files\WebSite"` including the double quotes.
3. The prompt should now look like this `C:\Program Files\WebSite> type bin\python z2.py -xx ...` where -xx is one of the options from the chart below.

### To start ZServer using the startup batch file

A file by the name of **start.bat** is in **C:\Program Files\WebSite\**. The contents are listed below. You can edit this file and place additional command line switched after the "-D" from the list above.

```
"C:\Program Files\WebSite\bin\python.exe" "C:\Program Files\WebSite\z2.py" -D  
%1 %2 %3 %4 %5 %6 %7 %8 %9
```

**Note: The above text should all be on one line, not split as it is. Another thing to be aware of is that Windows needs the double quotes to surround the names of the files because of the spaces in the file name.**

### To start ZServer using NT's service manager

If you installed Zope onto NT Server then you were prompted wether you would like to run Zope as an NT service. If you selected yes then you can use the the NT Service Manager to start and stop ZServer. To use the NT Service Manager:

1. Click on **Start->Settings->Control Panel**
2. Double click on the Services icon in the Control Panel
3. The NT Service Manager window will be displayed on the screen. Scroll down through the list of services and select the one labeled Zope.
4. Push the start or stop button
5. Configuring ZServer to use

## Using FastCGI With ZServer

FastCGI promises to eliminate the "Fork Tax"<sup>1</sup> that CGI scripts have to pay, because FastCGI is implemented as an add-on module for a wide range of web servers. The only caveat with this approach is that if you want to utilize FastCGI you will need to run ZServer in conjunction with your third party web server. If you are not running an additional web server then there is no need to use FastCGI. See the previous section "What is FastCGI" for more detail on what FastCGI is and why you would want to use it. Later in this chapter we will discuss how to run ZServer and FastCGI in conjunction with some of the popular web servers in use today.

There are two different methods that a web server can communicate with a FastCGI application: either through an INET port or a Unix Socket File. Unix socket files are slightly more efficient but the advantage of the INET Port is that the FastCGI application does not have to reside on the same machine as the web server.

### To use an INET Port

INET Ports can be used on both NT and Unix. All you need to specify is a port number with the -F xxxx where "xxxx" is the port number. You can chose any port number that is greater than 1024<sup>2</sup> and is not being used by another service.

The following two examples show how the start script should look like under Windows an Unix. This script instructs ZServer to listen for FastCGI requests on port 8999. The contents of your start script may be slightly different from the ones shown here.

#### *INET port start script example*

```
W | "C:\Program Files\WebSite\bin\python.exe" "C:\Program Files\WebSite\z2.py" -D
I | -F 8999
N
3
2
```

```
U | #!/bin/sh
N | reldir=`dirname $0`
I | PYTHONHOME=`cd $reldir; pwd`
X | export PYTHONHOME
   | exec /usr/bin/python \
   |     $PYTHONHOME/z2.py \
   |     -D \
   |     -F 8999
```

### To use FastCGI with a Unix socket file

This method only works on Unix. You can specify the path to a Unix socket file instead of using a port number. If the socket file does not exist then one will be created. You must have the appropriate privileges to read/write to the socket file and the location of the socket file should be on the local drive.

- 
1. The performance hit that is taken because a CGI script has to be started for a new for each web request.
  2. The first 1024 ports are reserved for well known services for example the default port for FTP is 21 where as the default port for HTTP is 80. On some operating systems only root or a person with supervisor privileges can run a service on a ports 1024 and below.

---

### Unix socket file start script example

```
U |#!/bin/sh
N |reldir=`dirname $0`
I |PYTHONHOME=`cd $reldir; pwd`
X |export PYTHONHOME
  |exec /usr/bin/python \
  |    $PYTHONHOME/z2.py \
  |    -D \
  |    -F /tmp/zope.soc
```

## Running Zope with third party web servers

Before ZServer, the only way to run Zope was in tandem with a third party web server<sup>1</sup>. This was done using PCGI as described earlier. With the inclusion of ZServer, the need to setup Zope with a third party web server is dwindling every day. However there are still a few scenarios that require the use of a third party server.

- You already have a third party web server running that you cannot take down and you cannot run ZServer on a different port. You might be behind a firewall that only allows port 80 through.
- Your existing web content is shared and/or modified over a network filesystem of some sort. You would like to protect all of your Zope content with SSL.
- You would like to use your existing web server to do advanced proxying so that you can do such things as advanced load balancing or use your server to map virtual hosts to different folders in the Zope hierarchy.
- You would like to run Zope on your ISP's machine.
- You are an ISP who would like to run an instance of Zope for each of your customers.

Whatever your reasons the next few sections will help you to get Zope running with the three most common web servers. We cover some of the above mentioned topics in the advanced sections as well.

---

1. This is actually a small fib, earlier versions of Zope were distributed with an application called ZopeHTTPServer. Although ZopeHTTPServer was ideal for the developer who wanted to quickly setup a Zope environment, its limited functionality kept most people from using it in a production environment.

---

---

**The number one problem encountered when running zope for the first time.**

Installing Zope should be as easy or as hard as it is to install a normal CGI script for your respective web server. Zope normally performs both authentication and authorization of users, however some web servers don't pass authentication information to CGI scripts. If you keep getting rejected when you try to access Zope through the web with the super manager user name and password, there is a good chance that your web server is not passing authentication information to Zope. Here we have instructions on how to get three of the most common web servers to pass authentication information to Zope.

## **Running Zope with IIS 4.0**

IIS is only available with Windows NT Server 4.0. Windows NT Server includes a free copy of Internet Information Server 2.0 which is often referred to as IIS. The following instructions assume that you have installed the Microsoft Options Pack which will upgrade your IIS to version 4.0. The option pack is available for free from <http://www.microsoft.com>.

**Note the rest of this section has not been completed yet.**

**IIS + PCGI**

**IIS + ZServer + PCGI**

**IIS + ZServer + FastCGI**

## Running Zope with Netscape

### Netscape + PCGI

### Netscape + ZServer + PCGI

### Netscape + ZServer + FastCGI

## Running Zope with Apache

### Apache + PCGI

### Apache + ZServer + PCGI

### Apache + ZServer + PCGI using mod\_pcg

### Apache + ZServer + FastCGI

Apache can be configured to use FastCGI support with the help of the mod\_fastcgi plug-in. It's available from <http://www.fastcgi.com>. Mod\_fastcgi can be a pain to setup for the first time because the documentation for mod\_fastcgi lacks examples. The good news is that understanding Apache's FastCGI configuration directives are a lot easier than figuring out how to use mod\_rewrite for PCGI; easier, once someone has explained their use which is what we are going to do in this section.

**NOTE: These instructions are only for the Unix version of Apache because mod\_fastcgi is not available for the Windows NT version of Apache at this time.**

The basic steps for configuring Apache and Zope to use FastCGI are:

- Download, Compile and Install mod\_fastcgi
- Configure ZServer to work with FastCGI
- Configure apache to pass requests via FastCGI to ZServer

### Download and install mod\_fastcgi

You can download the source to mod\_fastcgi from <http://www.fastcgi.com>. The latest stable version of mod\_fastcgi at the time of this writing is **2.2.2**. You'll need to either grab one of the latest snapshots or a newer version (if one exists) because a special undocumented feature has been added to mod\_fastcgi that Zope needs in order to handle authentication.

Download and uncompress the source archive and follow the instructions that are located in the INSTALL file. Once you have installed mod\_fastcgi you can move onto the next section.

### Configure ZServer to work with FastCGI

You need to add the `-F` directive to the ZServer start script. The `-F` directive instructs ZServer to listen on either a INET port or a Unix socket for FastCGI requests. Detailed instruction for configuring ZServer to use FastCGI can be found in the ZServer section.

### To Configure Apache to pass FastCGI request to ZServer via an INET Port

By now you should have the `mod_fastcgi` installed and working properly on your server. You can verify that the plug-in is functioning correctly by looking for the line that says "... `mod_fastcgi` configured -- resuming normal operations" in your Apache error log file. Refer to your server documentation if you are unsure of the location of the error log.

To get you started quickly, below is an example of what your `httpd.conf` file should look like. Insert these lines at the bottom of the file. Consult your server documentation if you are unsure where this file is located. The example below assumes that you told ZServer to listen for FastCGI request on INET port 8999.

```
FastCgiExternalServer /home/httpd/html/zope -host localhost:8999 -pass-header
Authorization

<Location /zope>
Options ExecCGI
SetHandler fastcgi-script
</Location>
```

Let's examine each part of the configuration syntax to understand what's going.

1. The first part of the first line is "`FastCgiExternalServer`" this directive tells Apache that we are going to be giving apache information about an external application that can handle FastCGI requests.
2. Immediately after that is "`/home/httpd/html/zope`" this points to a file in your document root directory. Whenever somebody tries to access this file from the web Apache will automatically redirect the request to the FastCGI application. This file can be empty but it must be readable and executable by the web server.
3. The next portion, "`-host localhost:8999`" instructs Apache to send the request to port 8999 of the machine that the web server is running on.
4. The final portion of the first line, by the way the instructions for the first four parts all have to be on the same line, reads "`-pass-header Authorization`" this argument instructs Apache to pass the authorization headers to the FastCGI application. Zope needs this so that it can preform it's own Authentication and Authorization.
5. The next four lines of text starting with "`<Location /zope>`" are mapping directives that instruct Apache that the requested URL, in this case "`/zope`" is to be handled by `mod_fastcgi`. The only thing you might want to change is "`/zope`" if you want to change the name of the place holder file mentioned in item 2.

### To Configure Apache to pass FastCGI request to ZServer via a Unix socket

If you wish to use a Unix socket file you'll need to add the "`FastCgiIpcDir directoy`" directive to the example in the above section. This directive tells Apache the directory where the socket files will be found. The only other thing you'll need to do is to remove the `-host localhost:8999` and replace it with `-socket file`, which should be the file name of the socket that ZServer is listening on. `mod_fastcgi` expects the socket file to be relative to the `FastCgiIpcDir`, not specifying this directive or including the full path to the socket file will keep `mod_fastcgi` from working.

Here is an example of what the configuration file should look like

```
FastCgiIpcDir /tmp

FastCgiExternalServer /home/httpd/html/zope -socket zope.sock -pass-header Authorization

<Location /zope>
Options ExecCGI
SetHandler fastcgi-script
</Location>
```

Let's examine this example.

1. The first directive "`FastCgiIpcDir /tmp`" instructs Apache to look in **/tmp** for the socket file named with the directive `-socket`. The Apache process must have the appropriate privileges to read and write to this socket file.
2. The next line is identical to the first line in the previous example except that we have replaced `-host localhost:8999` with `-socket zope.sock`. Apache combines this file name with the directory named using the `FastCgiIpcDir` directive to know that the full path to the socket file is **/tmp/zope.sock**.
3. The rest of the example is identical to the previous one.

