

---

# BlueShoes Installation:

## *get blueshoes running*

---

Andrej Arn  
Sam Blume



blueshoes

**version:**

1.4, updated 2006-06-14

**requirements:**

apache (1.3.x or 2) and php >= 4.2 are already installed on your system. (recommended is php 4.3, but version 5 works too. Betas of php6 have not been tested yet.)

you have linux, windows or whatever.

understanding: how to configure apache, add a dns or hosts entry

**pre-notes:**

it is wise to create similar environments everywhere you run your sites. dev machines, live servers etc. that will make your life easier!

**get the code**

*using the distributed package*

download it from [www.blueshoes.org](http://www.blueshoes.org)

unpack it on your disk. our recommended folder, and folder we use in this guide is:

`/usr/local/lib/php/blueshoes-4.6/` (whatever version you have, here 4.6).

now in the file system set a link named

`/usr/local/lib/php/blueshoes/` to the 4.6 folder.

from now on we'll use that path. if you download 4.7 in the future, put it in

`/usr/local/lib/php/blueshoes-4.7/` and just update the link to show to this dir.

if you're on an os that does not have such links, like windows,

make a copy. or forget that step, and just create the un-numbered path version. on windows your path will start with a drive letter, like `c:/usr/...`

if you cannot create that path, just put it elsewhere. it is recommended not to put the blueshoes library into the document root of your web server. but if you have to, well, do it.

you now have folders like this:

`/usr/local/lib/php/blueshoes/applications`

`/usr/local/lib/php/blueshoes/core`

...

note: linux only understands foreslashes as path separators, while windows understands both backslashes and foreslashes. so use foreslashes all the time.

*using cvs*

register as a developer and apply for a cvs account.

the cvs install manual is located here:

[http://www.blueshoes.org/manuals/Bs\\_InstallationCvs.manual.pdf](http://www.blueshoes.org/manuals/Bs_InstallationCvs.manual.pdf)

you already have a web site, or want to create a new one. let's update the apache configuration.

### hosts

let's say you own blueshoes.org and want to develop that site on your workstation. to make things easier, i recommend adding a hosts entry. you can do that in your dns server, or on windows you can just edit your hosts file:

it's called "hosts" with no file extension and can be found somewhere in the windows folder. winnt and win2k should have it in c:/winnt/system32/drivers/etc/.

add this line:

```
127.0.0.1 dev.www.blueshoes.org
```

save the file, and the new host should work. you may want to use your real ip address instead of 127.0.0.1.

if you have a real domain pointed to your box, that's ok too :-)

if you don't want to do all that, you'll have to use your ip address to access the dev site.

### create dirs for the website

now we need to put the website somewhere.

as example, we use the dir /web/ to put all the sites in. for the blueshoes site, we create a subdir, like

```
/web/www.blueshoes.org/
```

so if you have other domains, they look like:

```
/web/www.othersite.com/
```

now we create subdirs:

```
/web/www.blueshoes.org/online/
```

```
/web/www.blueshoes.org/logs/
```

```
/web/www.blueshoes.org/data/
```

```
/web/www.blueshoes.org/whatever/
```

your document root (the directory that gets hit when one enters your domain name into his browser) is "online", nothing else. so you can still have things that belong to this site, but are not accessible from the web. (i wish all isp's would set up the hosting accounts like this...)

the "online" dir could also be named "webroot", "web", "www", "htdocs" or whatever.

we call it online by convention. if you go with that, you need a bit less manual configuration.

note: even if we have created a hosts entry named "dev.www.blueshoes.org" and not just "www.blueshoes.org" i highly recommend to name your directory like the final domain name. because when you create that configuration on the web server, it's easier if the path matches.

**configure apache: httpd.conf**

let's add (or update) the site entry in your apache configuration. that can be in the httpd.conf, but maybe you have your sites configured in a file named srm.conf.

```
#blueshoes.org
<Directory "/web/www.blueshoes.org/online">
  Options FollowSymLinks Indexes
  AllowOverride All
  Order deny,allow
  Allow from all
</Directory>
```

feel free to change these options, like indexes etc. or to restrict the access from the outside world.

```
<VirtualHost *>
  DocumentRoot /web/www.blueshoes.org/online/
  ServerName www.blueshoes.org
  #add whatever serveralias you want
  ServerAlias dev.www.blueshoes.org

  #selfmade error docs wouldn't be bad.
  #the site blueshoes.org uses that, for example see http://www.blueshoes.org/foo/bar/
  #ErrorDocument 403 /err403.html
  #ErrorDocument 404 /err404.html

  #alias for coding and debugging
  Alias /_bsCore /usr/local/lib/php/blueshoes/core
  Alias /_bsPlugins /usr/local/lib/php/blueshoes/plugins
  Alias /_bsApplications /usr/local/lib/php/blueshoes/applications

  #alias for real usage
  Alias /_bsToolbox /usr/local/lib/php/blueshoes/toolbox
  Alias /_bsJavascript /usr/local/lib/php/blueshoes/javascript
  Alias /_bsImages /usr/local/lib/php/blueshoes/images
  Alias /_bsLib /usr/local/lib/php/blueshoes/lib
  Alias /_libDomapi C:/cvs/domapi_3rc2_Standard/src
</VirtualHost>
```

note: if you're on windows, you may have to specify the path for the aliases like "c:/usr/local/lib/php/blueshoes/toolbox", with the drive letter in front. there was a bug in apache 1.3.something that required this.

on a live box you may want to disable the first group of aliases for security reasons.

the 2<sup>nd</sup> part (alias for real usage) is absolutely needed. using this, you'll be able to hit files directly, eg [http://www.blueshoes.org/\\_bsImages/spacer.gif](http://www.blueshoes.org/_bsImages/spacer.gif)

if you are not allowed to edit the apache configuration, and cannot achieve the same results with htaccess files (does this work?), then you need to \*copy\* the directories into subdirs in your doc root, example:

copy /usr/local/lib/php/blueshoes/images to /web/www.blueshoes.org/online/\_bsImages.  
this is no fun, but works.

Make sure that `$_SERVER['DOCUMENT_ROOT']` ends with a slash. I believe that when your DocumentRoot definition in the VirtualHost directive has one, you are fine.

### **configure blueshoes for your site; global.conf.php**

grab the file “global.conf.php-dist” from the root dir of your unpacked blueshoes and put it into the “/web/www.blueshoes.org/” directory, so one above the webroot. if you’re on a server where you cannot break out of the webroot, put the file into your webroot. that sucks, but happens and with some hacks it can be supported. rename it to “global.conf.php”. open it, we need to do some changes. as you can see it’s a php file. no xml or ini file to parse, so it’s fast and simple.

if you want to use sessions, make sure the path specified in `$APP['sess']['path']` exists. by default that would be /web/www.blueshoes.org/session/. also set `$sessionType` to something, namely ‘simple’ or ‘file’.

if you want to use an automatic browser check as blueshoes.org does it, activate `$APP['browscap']['use']`.

set the vars `$APP['emailWebmaster']` and `$APP['emailSysop']`.

update the array `$APP['usedLanguages']`, specify the languages used on your site.

find the key `$APP['db']` and update that array. later down the file you can open up the default db connection. the advantage of doing it in the config file is that you don’t have to do it in your code anymore.

it is wise to create a database for each website. i recommend setting `$APP['db']['main']['name'] = “siteblueshoes”`; (or so, use the name of your site).

then create such a db:

create database siteblueshoes;

your db user needs full access to this db, and needs at least read access to the BsKb database (if you use the knowledge base).

remember: if you add a mysql db user, don’t forget to execute “flush privileges”.

### **kb**

the kb is distributed in packages on their own, and made available to registered developers and customers only. they include sql files you can import directly. they’re made for mysql, but it shouldn’t be hard to use them for any other rdbms.

now restart your apache, and blueshoes should be running.

having problems? read this file again. then go to <http://developer.blueshoes.org/forum/>.

**mod\_gzip**

If cpu isn't so much an issue on your server, and you want to save bandwidth and speed up your site, i recommend you install apache mod\_gzip.

it will compress the content you specify (php output, js files, static html files, whatever). browsers since 1998 are capable of reading compressed input, others are fed with uncompressed nourishment.

The url for this used to be:

[http://www.remotecomunications.com/apache/mod\\_gzip/](http://www.remotecomunications.com/apache/mod_gzip/)

But they seem to have lost the domain and a domain dealer got it. So use Google.

**php-accelerator**

also consider installing a php cache. you may want to read the file

[http://www.blueshoes.org/articles/Bs\\_Speed\\_Matters.article.pdf](http://www.blueshoes.org/articles/Bs_Speed_Matters.article.pdf)