



Server Administrator Manual

Blue Quartz Server Management Site Management BlueInO Personal Profile

Server Administrators
Network Services
Security
System Settings
Maintenance
Usage Information
Active Monitor
Solarspeed GUI
AV.Suite
Security Package
SPAM-Filter
MRTG Graphs
Server-Backup
OpenWebmail
Solarspeed PHP
License
logviewermenuentry

Configure your php.ini settings

php.ini security settings Expert mode

Register Globals:	Off
Safe Mode:	On
Safe Mode GID:	Off
Safe Mode include directory:	
Safe Mode exec directory:	
Safe Mode allowed envelope variables:	PHP_
Safe Mode protected envelope variables:	LD_LIBRARY_PATH
Open basedir:	
Disable functions:	
Disable classes:	

Save

? For maximum security you must set 'register_globals' to 'Off'. Otherwise it is easily possible that unsecurely programmed PHP scripts can be used to hack your server.



Solarspeed.net mod_php for CentOS + Blue Quartz



Introduction

The Solarspeed.net MOD_PHP is an updated PHP installation, which installes into a separate directory than the already present PHP-4.3.9. This allows you to run the old PHP-4.3.9 which the BlueQuartz interface requires and the new PHP side by side without any interference.

The AdmServ (the Apache webserver that powers the GUI interface) will continue to use the old PHP-4.3.9, while the public Apache webserver which serves your webpages will use the newer and more modern PHP version.

The new PHP will be installed into the directory **/home/solarspeed/mod_php/** and it comes with all the bells and whistles enabled that the stock PHP is missing. The new features include support for XML, PDF, cURL, DOM-XML, Mcrypt, Mbstring, MHash, Aspell/Pspell, ZIP and Zzip – as well as Zend Optimizer and IonCube support.

To remain as compatible as possible with the BlueQuartz base install the command line interpreter for PHP at **/usr/bin/php** has to remain unmodified. However, if you need to run PHP scripts through the more modern CLI that our PHP provides, then you find the CLI for that at the location **/home/solarspeed/mod_php/bin/php**

Just execute your scripts with the new CLI and they will be executed with the updated PHP interpreter instead. Example: **/home/solarspeed/mod_php/bin/php myscript.php**

Likewise our updated PHP uses a separate php.ini file. You find the new php.ini at this location:

/home/solarspeed/mod_php/etc/php.ini

You can edit this php.ini through the GUI interface. Additional .ini files placed into **/home/solarspeed/mod_php/etc/php.d/** will be included by the updated PHP if you provide them yourself.

To provide you with the most straightforward way on installing and using our MOD_PHP, the software is available as self installable PKG file, which you can install yourself. Either through the GUI interface, or from the command line.



PKG installation

When you have purchased the software and downloaded the PKG file to your workstation, you can install it on the server by logging in into the GUI interface as admin. Click on “BlueLinQ”, “Third Party Software” (“Third Party Updates” works as well) and then choose upload.



Once the PKG is installed, click on “Installed Software” and you should see it listed there.

Now it is time to activate the software:



MOD_PHP needs a license key for each server it runs on.

To get a license key, click on “Server Management” and you should see a new menu entry on the left side: “Solarspeed GUI”. Click on it to expand and you will see the entry “Solarspeed PHP”.

When you click on “Solarspeed PHP”, it will expand show another menu entry below it, labeled “License”.

Click on “License” and you will see the menu shown on the next page of this document:



Solarspeed.net mod_php for CentOS + Blue Quartz



Product License

License management:

Customer Name: Michael Stauber

Customer Email: mstauber@solarspeed.net

Company Name: Stauber Multimedia Des

Server Key: ----- SERVER ADAPTER DATA -----
S71T7X23RGKjr1HOKJ/7ChHEMuDqBxb5
u802ONvRh3kAmzg6Fo/1D11uKK3f8xvd
vT54tS+QHAA4p0D6c0irTTs1f24a7K1a
68mX5B5G0Zg1c6wzBuAJGhrsazFa2R1x
VjoJ9FD1TA7/P/NOuTOLT80Gw+g/Ih9P
c/==

Product Key: ----- LICENSE FILE DATA -----
74029603rnhsPbUm5/jAQ7HuGnKb2ORe
yL1f2Iw3hxBr1y7G8oRu1Mp/e70JrXJ3
6uTtAy9B0ghmSEvBvYLenb8gSesSvw4G
O+suvEas7/YA6S0d1b9P1Y9cR6oaJWB1
P1EEyysfkMox7xWyCYBcNOuOpAdv1gn
sprU3yqLncLq2ydCDTGvCBwnShpFC3VS
pp7FN02roIqp9sAqo39zj/hK3Q+eXCFw
ptOWLaa8TcnRUV9s+7f1kGd8VTzI9fHX

License key generation and installation:
This software is under copyright of Stauber Multimedia Design (<http://www.solarspeed.net>).
A license is required for the software to work. The license can be obtained by purchasing the software in our online store at <http://www.solarspeed.net>.
If you have purchased the software and have no License Key yet, then first fill in your name, email address and company data. Then click on the "Save" button. Once that is done, click on the "Email Key Request" button to email your "Server Key" to keyrequest@solarspeed.net. When we have received and validated your key request, we will email you a keyfile which you can post into the field "Product Key". Once you have posted the key click on the "Save" button again. As soon as a valid key has been entered, the product is unlocked and ready to run.

For more information please refer to the PDF manual.

Email Key Request

Manage your OpenWebmail License.

You need to fill in the fields “Customer Name”, “Customer Email” and “Customer Company”.

The box “Server Key” will already be filled out. It contains encrypted data which is generated by the environment of your server (host name, domains, IP addresses, MAC addresses, system environment, etc.) and helps us to bind the software to a specific server.

If you already have a “Product Key”, you can directly enter it into the “Product Key” field and press “Save”.

If you do not have a “Product Key” yet, click on “Save” first (to store your name, email address and company name). Then click on “Email Key Request”.



Solarspeed.net mod_php for CentOS + Blue Quartz



Once you do that, an email is sent to Solarspeed.net which contains your name, email address, company name and the encrypted “Server Key”. We then generate a “Product Key” which we will email to you as soon as possible.

Please keep a copy of this key on file, as you may need it again if you ever need to do an OS restore and reinstall of the MOD_PHP PKG.

Once you have the key, enter the key into the “Product Key” fields and press “Save”. Once you do that, the MOD_PHP RPM will be installed and your Apache webserver will be restarted. Afterwards all PHP enabled sites will be parsed by the new PHP, while the GUI continues to run with the old onboard PHP-4.3.9.

Please note:

The “Product Key” does not expire. But it will not work for future versions of our MOD_PHP other than the version that you bought. If you want to run MOD_PHP on more than one server, then you need to purchase multiple Product Keys. Bulk deals are available.

PHP Security - IMPORTANT!

Configure your php.ini settings	
	php.ini security settings Expert mode
Register Globals:	Off <input type="button" value="v"/>
Safe Mode:	On <input type="button" value="v"/>
Safe Mode GID:	Off <input type="button" value="v"/>
Safe Mode include directory:	<input type="text"/>
Safe Mode exec directory:	<input type="text"/>
Safe Mode allowed envelope variables:	PHP_ <input type="text"/>
Safe Mode protected envelope variables:	LD_LIBRARY_PATH <input type="text"/>
Open basedir:	<input type="text"/>
Disable functions:	<input type="text"/>
Disable classes:	<input type="text"/>

Save

This screen (reachable through “Solarspeed GUI” / “Solarspeed PHP”) will allow you to configure the most important and security relevant switches in php.ini.

By clicking on “Expert mode” you can also edit the entire php.ini file through an text editor.



Solarspeed.net mod_php for CentOS + Blue Quartz



The way the switches are set during default installation presents the most secure fashion and **you should not modify these switches – unless you want to risk getting hacked.** Or unless you want to set it even tighter, like by disabling functions or classes and/or setting an Open Basedir restriction.

Today the two most common causes for compromised BlueQuartz servers are the following two issues:

- a) Easily guessable or brute-forceable admin and/or user passwords.
- b) Installation of vulnerable PHP scripts

When a PHP script is executed on your server, it runs under the user ID of the owner of the script (or user “apache”) and the group ID of the site in question that the script belongs to. If unset this also defaults to “apache”.

PHP is a very powerful scripting language and unless you put some security measures in, it can be used to run unwanted code and operations on your server. As mentioned, PHP usually runs as unprivileged user, but there are ways possible how someone can gain root access by exploiting certain vulnerabilities in installed software to gain root access.

To prevent this three ways are imaginable:

- a) Disable PHP for all sites. This usually is not an option.
- b) Make sure only trusted and well programmed PHP scripts are installed on your server. While this would work in a perfect world, usually you have little control over which scripts a client installs on your server. Or a script that you deemed secure gets exploited nonetheless, because it contained bad code that you were not aware of.
- c) Limit what PHP scripts can do. Lock it down tightly, so that most malicious code cannot be run. This document will explain some steps to you how you can do this. Additionally, the basic configuration of our MOD_PHP already provides you with a much better security than the stock PHP.

Register Globals (default = Off):

By default, PHP makes all environment and server variables, all cookies, and all GET and POST variables globally accessible by name. This helps novice PHP programmers greatly -- they do not have to figure out how to retrieve this external data. But it is a dangerous practice in that important settings can accidentally be changed very easily and transparently. Worse yet, there isn't much to stop an attacker from submitting a form field with the same tag name as that of a variable containing some file name, for instance. The configuration directive `register_globals` controls this aspect of PHP's behavior, and it seems appropriate to turn it off.

Otherwise an attacker can just call your scripts in a web browser and supply his own values for your variables and his own code for execution through a modified URL string. All he needs to know is your variable names – and if you are using a popular script like phpBB, Joomla!, Mambo or there like, then you risk getting exploited, defaced or hacked is multiplied by several magnitudes.



Safe Mode (default = On):

PHP can be set up so that it executes scripts in a restricted environment to decrease the amount of damage that can be inflicted by insecure programs. This modus operandi is called "safe mode". The configuration directive `safe_mode` in `php.ini` turns safe mode on and off.

safe_mode_exec_dir (default: not set):

The `safe_mode_exec_dir` directive specifies a directory from which scripts can be loaded. PHP will not execute a script if it is not in this directory. Furthermore, PHP will not let a script call another program that is not in this directory. This way, even if there is a security hole in the script that allows attackers to run arbitrary commands on the script, they will be limited to whatever is in the safe mode executable directory.

safe_mode_allowed_env_vars (default: not set):

The field `safe_mode_allowed_env_vars` contains a list of prefixes that identify the names of the environment variables the user is allowed to change. Thus, any environment variable whose name begins with something not listed in `safe_mode_allowed_env_vars` cannot be altered from within a PHP script. The default list consists of the prefix "PHP_" only. As we have seen, some of the PHP_ variables also contain sensitive information, so this restriction does not always solve the problem completely.

safe_mode_protected_env_vars (default: not set):

Another cognate configuration setting is `safe_mode_protected_env_vars`. The list given to this directive specifies names of environment variables that the user is not allowed to modify. The protected variables cannot be altered even if they are also present in the `safe_mode_allowed_env_vars` list. By default, the only protected variable is `$LD_LIBRARY_PATH`.

For increased safety, it would be best to use both settings as complementary, placing as many "endangered" environment variables in the directive `safe_mode_protected_env_vars` as possible. As a general rule, if it is not absolutely necessary for scripts to be able to alter a variable, protect it. By all means, try to protect `$PATH` and friends. If this is not an acceptable solution, regard all unprotected variables with distrust and handle them with care.

open_basedir (default: not set):

The directive `open_basedir` specifies the root of a directory tree outside of which scripts are not allowed to open files. That is, a script can not use `fopen()`, for instance, on a file that is not under that directory tree. By default, `open_basedir` is empty and PHP allows files to be opened anywhere (provided that the script has appropriate access permissions). On BlueQuartz it would be a good idea to set `open_basedir` to `/home`. That way scripts cannot access system files located in `/etc` or other sensitive areas.



Solarspeed.net mod_php for CentOS + Blue Quartz



disable_functions (default: not set):

Another restrictive directive is `disable_functions`. This lists comma-separated names of functions that PHP will just ignore. Putting `dl()` on this list is another way to forbid dynamic loading. It is probably good to put `phpinfo()` there, because it gives out so much information about the script and the host. For even tighter security, you can disable `mail()`, `system()`, and `friends`, even `include()`. Of course this also limits the functionality of the script quite severely. Generally, it is a good idea to disable functions that have the potential to do damage and that the scripts can do without.

To read more on the topic of securing PHP I recommend the following resources:

On the Security of PHP, Part 1 (Jordan Dimov)

http://softwaredev.earthweb.com/script/article/0,,12063_918141,00.html

On the Security of PHP, Part 2 (Jordan Dimov)

<http://www.developer.com/lang/article.php/922871>

Using Register Globals:

http://us3.php.net/register_globals



Copyright, License and Disclaimer

Copyright:

This software and its documentation is © 2005-2006 by Michael Stauber of Stauber Multimedia Design (<http://www.solarspeed.net>). Redistribution or resale of the software without explicit permission is not allowed. The documentation may be copied freely, as long as it is distributed in full and without any changes or modifications.

License:

The software “Solarspeed.net MOD_PHP” can be purchased through the Solarspeed.net online store at this URL:

http://www.solarspeed.net/cart.php?target=product&product_id=16159&category_id=250

The price for a single server license is 50,- EUR. It is also available at discounted price together with our [PHP, MySQL and phpMyAdmin bundle](#). The PKG of the software will be provided upon purchase for self install. Install of the software through a Solarspeed.net technician is available free of charge.

After the installation a “Product Key” (a.k.a. License) is required to unlock the software. The “Product Key” can be requested through the GUI interface and will be delivered by email. If the software is installed by one of our technicians, then a key will immediately be installed with the software.

The key is non transferable and binds the software to the server. In case of a hardware defect and/or replacement of the server or change of ISPs a new “Product Key” may be requested for the new server. Usually this key for the new server will be provided free of charge, but if we suspect fraud, we may reserve the right to charge for the new Product Key.

Disclaimer:

We have tested this software carefully under various circumstances which mimic the most common scenarios that one might encounter during backing up and restoring a BlueQuartz server appliance. Yet - complex program code is seldom fully free of errors. So while the software might have worked 100% fine for us and for most of our clients, it can be that it still contains errors which you might eventually run into.

If you encounter problems with the software, please contact us, so that we can resolve this issues: <http://www.solarspeed.net/cart.php?target=help&mode=contactus>