FAQ: Frequently Asked Questions

General Information

This section holds the most general questions about PHP: what it is and what it does.

What is PHP?

From the preface of the manual:

PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

What does PHP stand for?

PHP stands for *PHP: Hypertext Preprocessor*. This confuses many people because the first word of the acronym is the acronym. This type of acronym is called a recursive acronym. The curious can visit » Free On-Line Dictionary of Computing for more information on recursive acronyms.

What is the relation between the versions?

PHP/FI 2.0 is an early and no longer supported version of PHP. PHP 3 is the successor to PHP/FI 2.0 and is a lot nicer. PHP 4 is the current generation of PHP, which uses the » Zend engine under the hood. PHP 5 uses the Zend engine 2 which, among other things, offers many additional OOP features.

Can I run several versions of PHP at the same time?

Yes. See the *INSTALL* file that is included in the PHP source distribution.

What are the differences between PHP 3 and PHP 4?

Here's a list of some of the more important new features:

- Extended API module
- Generalized build process under Unix
- Generic web server interface that also supports multi-threaded web servers
- Improved syntax highlighter
- Native HTTP session support
- Output buffering support
- More powerful configuration system
- Reference counting

Please see the » What's new in PHP 4 overview for a detailed explanation of these features and more.

I think I found a bug! Who should I tell?

You should go to the PHP Bug Database and make sure the bug isn't a known bug. If you don't see it in the database, use the reporting form to report the bug. It is important to use the bug database instead of just sending an email to one of the mailing lists because the bug will have a tracking number assigned and it will then be possible for you to go back later and check on the status of the bug. The bug database can be found at » http://bugs.php.net/.

Mailing lists

This section holds questions about how to get in touch with the PHP community. The best way is the mailing lists.

Are there any PHP mailing lists?

Of course! There are many mailing lists for several subjects. A whole list of mailing lists can be found on our » Support page.

The most general mailing list is *php-general*. To subscribe, send mail to » php-general-subscribe@lists.php.net. You don't need to include anything special in the subject or body of the message. To unsubscribe, send mail to » php-general-unsubscribe@lists.php.net.

You can also subscribe and unsubscribe using the web interface on our » Support page.

Are there any other communities?

There are countless of them around the world. We have links for example to some IRC servers and foreign language mailing lists on our » Support page.

Help! I can't seem to subscribe/unsubscribe to/from one of the mailing lists!

If you have problems subscribing to or unsubscribing from the php-general mailing list, it may be because the mailing list software can't figure out the correct mailing address to use. If your email address was *joeblow@example.com*, you can send your subscription request to *php-general-subscribe-joeblow=example.com@lists.php.net*, or your unsubscription request to *php-general-unsubscribe-joeblow=example.com@lists.php.net*. Use similar addresses for the other mailing lists.

Is there an archive of the mailing lists anywhere?

Yes, you will find a list of archive sites on the » Support page. The mailing list articles are also archived as news messages. You can access the news server at » news://news.php.net/ with a news client. There is also an experimental web interface for the news server at » http://news.php.net/

What can I ask the mailing list?

Since PHP is growing more and more popular by the day the traffic has increased on the php-general mailing list and as of now the list gets about 150 to 200 posts a day. Because of this it is in everyone's interest that you use the list as a last resort when you have looked everywhere else.

Before you post to the list please have a look in this FAQ and the manual to see if you can find the help there. If there is nothing to be found there try out the mailing list archives (see above). If you're having problem with installing or configuring PHP please read through all included documentation and README's. If you still can't find any information that helps you out you're more than welcome to use the mailing list.

Before asking questions, you may want to read the paper on » How To Ask Questions The Smart Way as this is a good idea for everyone.

What information should I include when posting to the mailing list?

Posts like "I can't get PHP up and running! Help me! What is wrong?" are of absolutely no use to anyone. If you're having problems getting PHP up and running you must include what operating system you are running on, what version of PHP you're trying to set up, how you got it (pre-compiled, CVS, RPMs and so on), what you have done so far, where you got stuck and the exact error message.

This goes for any other problem as well. You have to include information on what you have done, where you got stuck, what you're trying to do and, if applicable, exact error messages. If you're having problems with your source code you need to include the part of the code that isn't working. Do not include more code than necessary though! It makes the post hard to read and a lot of people might just skip it all together because of this. If you're unsure about how much information to include in the mail it's better that you include to much than to little.

Another important thing to remember is to summarize your problem on the subject line. A subject like "HELP MEEEE!!!" or "What is the problem here?" will be ignored by the majority of the readers.

And lastly, you're encouraged to read the paper on » How To Ask Questions The Smart Way as this will be a great help for everyone, especially yourself.

Obtaining PHP

This section has details about PHP download locations, and OS issues.

Where can I obtain PHP?

You can download PHP from any of the members of the PHP network of sites. These can be found at » http://www.php.net/. You can also use anonymous CVS to get the absolute latest version of the source. For more information, go to » http://www.php.net/anoncvs.php.

Are pre-compiled binary versions available?

We only distribute precompiled binaries for Windows systems, as we are not able to compile PHP for every major Linux/Unix platform with every extension combination. Also note, that many Linux distributions come with PHP built in these days. Windows binaries can be downloaded from our » Downloads page, for Linux binaries, please visit your distribution's website.

Where can I get libraries needed to compile some of the optional PHP extensions?

Note
Those marked with * are not thread-safe libraries, and should not be used with PHP as a server module in the multi-threaded Windows web servers (IIS, Netscape). This does not matter in Unix environments, yet.

- » LDAP (Unix).
- » LDAP (Unix/Win): Mozilla Directory (LDAP) SDK
- » free LDAP server.
- » Berkeley DB2 (Unix/Win): http://www.sleepycat.com/.

- » SNMP* (Unix):.
- » GD* (Unix/Win).
- » mSQL* (Unix).
- » PostgreSQL (Unix).
- » IMAP* (Win/Unix).
- » Sybase-CT* (Linux, libc5): Available locally.
- » FreeType (libttf):.
- » ZLib (Unix/Win32).
- » expat XML parser (Unix/Win32).
- » PDFLib.
- » mcrypt.
- » mhash.
- » t1lib.
- » dmalloc.
- » aspell.
- » readline.

How do I get these libraries to work?

You will need to follow instructions provided with the library. Some of these libraries are detected automatically when you run the 'configure' script of PHP (such as the GD library), and others you will have to enable using ' *--with-EXTENSION* ' options to ' *configure* '. Run ' *configure --help* ' for a listing of these.

I got the latest version of the PHP source code from the CVS repository on my Windows machine, what do I need to compile it?

First, you will need Microsoft Visual C++ v6 (v5 may do it also, but we do it with v6), and you will need some support files. See the manual section about building PHP from source on Windows.

Where do I find the Browser Capabilities File?

You can find a *browscap.ini* file at » http://browsers.garykeith.com/downloads.asp.

Database issues

This section holds common questions about relation between PHP and databases. Yes, PHP can access virtually any database available today.

I heard it's possible to access Microsoft SQL Server from PHP. How?

On Windows machines, you can simply use the included ODBC support and the correct ODBC driver.

On Unix machines, you can use the Sybase-CT driver to access Microsoft SQL Servers because they are (at least mostly) protocol-compatible. Sybase has made a » free version of the necessary libraries for Linux systems. For other Unix operating systems, you need to contact Sybase for the correct libraries. Also see the answer to the next question.

Can I access Microsoft Access databases?

Yes. You already have all the tools you need if you are running entirely under Windows 9x/Me, or NT/2000, where you can use ODBC and Microsoft's ODBC drivers for Microsoft Access databases.

If you are running PHP on a Unix box and want to talk to MS Access on a Windows box you will need Unix ODBC drivers. » OpenLink Software has Unix-based ODBC drivers that can do this.

Another alternative is to use an SQL server that has Windows ODBC drivers and use that to store the data, which you can then access from Microsoft Access (using ODBC) and PHP (using the built in drivers), or to use an intermediary file format that Access and PHP both understand, such as flat files or dBase databases. On this point Tim Hayes from OpenLink software writes:

> Using another database as an intermediary is not a good idea, when you can use ODBC from PHP straight to your database - i.e. with OpenLink's drivers. If you do need to use an intermediary file format, OpenLink have now released Virtuoso (a virtual database engine) for NT, Linux and other Unix platforms. Please visit our » website for a free download.

One option that has proved successful is to use MySQL and its MyODBC drivers on Windows and synchronizing the databases. Steve Lawrence writes:

- Install MySQL on your platform according to instructions with MySQL. Latest available from » http://www.mysql.com/ No special configuration required except when you set up a database, and configure the user account, you should put % in the host field, or the host name of the Windows computer you wish to access MySQL with. Make a note of your server name, username, and password.
- Download the MyODBC for Windows driver from the MySQL site. Install it on your Windows machine. You can test the operation with the utilities included with this program.
- Create a user or system dsn in your ODBC administrator, located in the control panel. Make up a dsn name, enter your hostname, user name, password, port, etc for you MySQL database configured in step 1.
- Install Access with a full install, this makes sure you get the proper add-ins... at the least you will need ODBC support and the linked table manager.
- Now the fun part! Create a new access database. In the table window right click and select Link Tables, or under the file menu option, select Get External Data and then Link Tables. When the file browser box comes up, select files of type: ODBC. Select System dsn and the name of your dsn created in step 3. Select the table to link, press OK, and presto! You can now open the table and add/delete/edit data on your MySQL server! You can also build queries, import/export tables to MySQL, build forms and reports, etc.

Tips and Tricks:

- You can construct your tables in Access and export them to MySQL, then link them back in. That makes table creation quick.
- When creating tables in Access, you must have a primary key defined in order to have write access to the table in access. Make sure you create a primary key in MySQL before linking in access
- If you change a table in MySQL, you have to re-link it in Access. Go to tools>add-ins>linked table manager, cruise to your ODBC DSN, and select the table to re-link from there. you can also move your dsn source around there, just hit the always prompt for new location checkbox before pressing OK.

PHP 5 no longer bundles MySQL client libraries, what does this mean to me? Can I still use MySQL with PHP? I try to use MySQL and get "function undefined" errors, what gives?

Yes. There will always be MySQL support in PHP of one kind or another. The only change in PHP 5 is that we are no longer bundling the client library itself. Some reasons in no particular order:

- Most systems these days already have the client library installed.
- Given the above, having multiple versions of the library can get messy. For example, if you link mod_auth_mysql against one version and PHP against another, and then enable both in Apache, you get a nice fat crash. Also, the bundled library didn't always play well with the installed server version. The most obvious symptom of this being disagreement over where to find the mysql.socket Unix domain socket file.
- Maintenance was somewhat lax and it was falling further and further behind the released version.
- Future versions of the library are under the GPL and thus we don't have an upgrade path since we cannot bundle a GPL'ed library in a BSD/Apache-style licensed project. A clean break in PHP 5 seemed like the best option.

This won't actually affect that many people. Unix users, at least the ones who know what they are doing, tend to always build PHP against their system's libmyqlclient library simply by adding the *--with-mysql=/usr* option when building PHP. Windows users may enable the extension *php_mysql.dll* inside *php.ini*. For more details, see the MySQL Reference for installation instructions. Also, be sure *libmysql.dll* is available to the systems PATH. For more details on how, read the FAQ on setting up the Windows systems PATH. Because *libmysql.dll* (and many other PHP related files) exist in the PHP folder, you'll want to add the PHP folder to your systems PATH.

After installing shared MySQL support, Apache dumps core as soon as libphp4.so is loaded. Can this be fixed?

If your MySQL libs are linked against pthreads this will happen. Check using ldd. If they are, grab the MySQL tarball and compile from source, or recompile from the source rpm and remove the switch in the spec file that turns on the threaded client code. Either of these suggestions will fix this. Then recompile PHP with the new MySQL libs.

Why do I get an error that looks something like this: "Warning: 0 is not a MySQL result index in <file> on line <x>" or "Warning: Supplied argument is not a valid MySQL result resource in <file> on line <x>"?

You are trying to use a result identifier that is 0. The 0 indicates that your query failed for some reason. You need to check for errors after submitting a query and before you attempt to use the returned result identifier. The proper way to do this is with code similar to the following:

```
<?php

$result = mysql_query("SELECT * FROM tables_priv");
```

```
if (!$result) {
   echo mysql_error();
   exit;
}
?>
```
or
```
<?php

$result = mysql_query("SELECT * FROM tables_priv")
   or die("Bad query: " . mysql_error());
?>
```

Installation

This section holds common questions about the way to install PHP. PHP is available for almost any OS (except maybe for MacOS before OSX), and almost any web server.

To install PHP, follow the instructions in Installation and Configuration.

Why shouldn't I use Apache2 with a threaded MPM in a production environment?

PHP is glue. It is the glue used to build cool web applications by sticking dozens of 3rd-party libraries together and making it all appear as one coherent entity through an intuitive and easy to learn language interface. The flexibility and power of PHP relies on the stability and robustness of the underlying platform. It needs a working OS, a working web server and working 3rd-party libraries to glue together. When any of these stop working PHP needs ways to identify the problems and fix them quickly. When you make the underlying framework more complex by not having completely separate execution threads, completely separate memory segments and a strong sandbox for each request to play in, feet of clay are introduced into PHP's system.

If you feel you have to use a threaded MPM, look at a FastCGI configuration where PHP is running in its own memory space.

And finally, this warning against using a threaded MPM is not as strong for Windows systems because most libraries on that platform tend to be threadsafe.

Unix/Windows: Where should my *php.ini* file be located?

By default on Unix it should be in */usr/local/lib* which is *<install-path>/lib*. Most people will want to change this at compile-time with the --with-config-file-path flag. You would, for example, set it with something like:

```
--with-config-file-path=/etc
```

And then you would copy *php.ini-dist* from the distribution to */etc/php.ini* and edit it to make any local changes you want.

```
--with-config-file-scan-dir=PATH
```

On Windows the default path for the *php.ini* file is the Windows directory. If you're using the Apache webserver, *php.ini* is first searched in the Apaches install directory, e.g. *c:\program files\apache group\apache*. This way you can have different *php.ini* files for different versions of Apache on the same machine.

See also the chapter about the configuration file.

Unix: I installed PHP, but every time I load a document, I get the message 'Document Contains No Data'! What's going on here?

This probably means that PHP is having some sort of problem and is core-dumping. Look in your server error log to see if this is the case, and then try to reproduce the problem with a small test case. If you know how to use 'gdb', it is very helpful when you can provide a backtrace with your bug report to help the developers pinpoint the problem. If you are using PHP as an Apache module try something like:

- Stop your httpd processes
- gdb httpd
- Stop your httpd processes
- > run -X -f /path/to/httpd.conf
- Then fetch the URL causing the problem with your browser
- > run -X -f /path/to/httpd.conf
- If you are getting a core dump, gdb should inform you of this now
- type: bt
- You should include your backtrace in your bug report. This should be submitted to » http://bugs.php.net/

If your script uses the regular expression functions (ereg() and friends), you should make sure that you compiled PHP and Apache with the same regular expression package. This should happen automatically with PHP and Apache 1.3.x

Unix: I installed PHP using RPMS, but Apache isn't processing the PHP pages! What's going on here?

Assuming you installed both Apache and PHP from RPM packages, you need to uncomment or add some or all of the following lines in your *httpd.conf* file:

```
# Extra Modules
AddModule mod_php.c
AddModule mod_php.c
AddModule mod_perl.c

# Extra Modules
LoadModule php_module         modules/mod_php.so
LoadModule php5_module        modules/libphp5.so     # for PHP 5
```

```
LoadModule perl_module        modules/libperl.so
```
And add:
```
AddType application/x-httpd-php .php
```
... to the global properties, or to the properties of the VirtualDomain you want to have PHP support added to.

Unix: I patched Apache with the FrontPage extensions patch, and suddenly PHP stopped working. Is PHP incompatible with the Apache FrontPage extensions?

No, PHP works fine with the FrontPage extensions. The problem is that the FrontPage patch modifies several Apache structures, that PHP relies on. Recompiling PHP (using 'make clean ; make') after the FP patch is applied would solve the problem.

Unix/Windows: I have installed PHP, but when I try to access a PHP script file via my browser, I get a blank screen.

Do a 'view source' in the web browser and you will probably find that you can see the source code of your PHP script. This means that the web server did not send the script to PHP for interpretation. Something is wrong with the server configuration - double check the server configuration against the PHP installation instructions.

Unix/Windows: I have installed PHP, but when try to access a PHP script file via my browser, I get a server 500 error.

Something went wrong when the server tried to run PHP. To get to see a sensible error message, from the command line, change to the directory containing the PHP executable (*php.exe* on Windows) and run *php -i*. If PHP has any problems running, then a suitable error message will be displayed which will give you a clue as to what needs to be done next. If you get a screen full of HTML codes (the output of the phpinfo() function) then PHP is working, and your problem may be related to your server configuration which you should double check.

Some operating systems: I have installed PHP without errors, but when I try to start apache I get undefined symbol errors:

```
[mybox:user /src/php4] root# apachectl configtest
apachectl: /usr/local/apache/bin/httpd Undefined symbols:
 _compress
```

```
_uncompress
```

This has actually nothing to do with PHP, but with the MySQL client libraries. Some need *--with-zlib*, others do not. This is also covered in the MySQL FAQ.

Windows: I have installed PHP, but when I to access a PHP script file via my browser, I get the error:

```
cgi error:
The specified CGI application misbehaved by not
returning a complete set of HTTP headers.
The headers it did return are:
```

This error message means that PHP failed to output anything at all. To get to see a sensible error message, from the command line, change to the directory containing the PHP executable (*php.exe* on Windows) and run *php -i*. If PHP has any problems running, then a suitable error message will be displayed which will give you a clue as to what needs to be done next. If you get a screen full of HTML codes (the output of the phpinfo() function) then PHP is working.

Once PHP is working at the command line, try accessing the script via the browser again. If it still fails then it could be one of the following:

- File permissions on your PHP script, *php.exe*, *php4ts.dll*, *php.ini* or any PHP extensions you are trying to load are such that the anonymous internet user *ISUR_<machinename>* cannot access them.
- The script file does not exist (or possibly isn't where you think it is relative to your web root directory). Note that for IIS you can trap this error by ticking the 'check file exists' box when setting up the script mappings in the Internet Services Manager. If a script file does not exist then the server will return a 404 error instead. There is also the additional benefit that IIS will do any authentication required for you based on the NTLanMan permissions on your script file.

Windows: I've followed all the instructions, but still can't get PHP and IIS to work together!

Make sure any user who needs to run a PHP script has the rights to run *php.exe* ! IIS uses an anonymous user which is added at the time IIS is installed. This user needs

rights to *php.exe*. Also, any authenticated user will also need rights to execute *php.exe* . And for IIS4 you need to tell it that PHP is a script engine. Also, you will want to read this faq.

When running PHP as CGI with IIS, PWS, OmniHTTPD or Xitami, I get the following error: *Security Alert! PHP CGI cannot be accessed directly.***.**

You must set the cgi.force_redirect directive to *0*. It defaults to *1* so be sure the directive isn't commented out (with a*;*). Like all directives, this is set in *php.ini*

Because the default is *1*, it's critical that you're 100% sure that the correct *php.ini* file is being read. Read this faq for details.

How do I know if my *php.ini* is being found and read? It seems like it isn't as my changes aren't being implemented.

To be sure your *php.ini* is being read by PHP, make a call to phpinfo(). Near the top, there will be a listing called *Configuration File (php.ini)*. This will tell you where PHP is looking for *php.ini* and whether or not it's being read. If just a directory PATH exists, then it's not being read, and you should put your *php.ini* in that directory. If *php.ini* is included within the PATH, it is being read.

If *php.ini* is being read and you're running PHP as a module, then be sure to restart your web server after making changes to *php.ini*

See also php_ini_loaded_file().

How do I add my PHP directory to the *PATH* on Windows?

On Windows NT, 2000, XP and 2003:

- Go to Control Panel and open the System icon (Start -> Settings -> Control Panel -> System, or just Start -> Control Panel -> System for Windows XP/2003)
- Go to the Advanced tab
- Click on the 'Environment Variables' button
- Look into the 'System Variables' pane
- Find the Path entry (you may need to scroll to find it)

- Double click on the Path entry
- Enter your PHP directory at the end, including ';' before (e.g.*;C:\php*)
- Press OK and restart your computer

On Windows 98/Me you need to edit the *autoexec.bat* file:

- Open the Notepad (Start -> Run and enter notepad)
- Open the *C:\autoexec.bat* file
- Locate the line with PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;..... and add: *;C:\php* to the end of the line
- Save the file and restart your computer

Note
Be sure to reboot after following the steps above to ensure that the *PATH* changes are applied.

The PHP manual used to promote the copying of files into the Windows system directory, this is because this directory (*C:\Windows*, *C:\WINNT*, etc.) is by default in the systems PATH. Copying files into the Windows system directory has long since been deprecated and may cause problems.

How do I make the *php.ini* file available to PHP on windows?

There are several ways of doing this. If you are using Apache, read their installation specific instructions (Apache 1, Apache 2), otherwise you must set the *PHPRC* environment variable:

On Windows NT, 2000, XP and 2003:

- Go to Control Panel and open the System icon (Start -> Settings -> Control Panel -> System, or just Start -> Control Panel -> System for Windows XP/2003)
- Go to the Advanced tab
- Click on the 'Environment Variables' button
- Look into the 'System variables' pane
- Click on 'New' and enter 'PHPRC' as the variable name and the directory where *php.ini* is located as the variable value (e.g. *C:\php*)
- Press OK and restart your computer

On Windows 98/Me you need to edit the *autoexec.bat* file:

- Open the Notepad (Start -> Run and enter notepad)
- Open the *C:\autoexec.bat* file
- Add a new line to the end of the file: *set PHPRC=C:\php* (replace *C:\php* with the directory where *php.ini* is located). Please note that the path cannot contain spaces. For instance, if you have installed PHP in *C:\Program Files\PHP*, you would enter *C:\PROGRA~1\PHP* instead.
- Save the file and restart your computer

Is it possible to use Apache content negotiation (MultiViews option) with PHP?

If links to PHP files include extension, everything works perfect. This FAQ is only for the case when links to PHP files don't include extension and you want to use content negotiation to choose PHP files from URL with no extension. In this case, replace the line *AddType application/x-httpd-php .php* with:

```
# PHP 4
AddHandler php-script php
AddType text/html php

# PHP 5
AddHandler php5-script php
AddType text/html php
```

This solution doesn't work for Apache 1 as PHP module doesn't catch *php-script*.

Is PHP limited to process GET and POST request methods only?

No, it is possible to handle any request method, e.g. CONNECT. Proper response status can be sent with header(). If only GET and POST methods should be handled, it can be achieved with this Apache configuration:

```
<LimitExcept GET POST>
Deny from all
</LimitExcept>
```

Build Problems

This section gathers most common errors that occur at build time.

I got the latest version of PHP using the anonymous CVS service, but there's no configure script!

You have to have the GNU autoconf package installed so you can generate the configure script from *configure.in*. Just run*./buildconf* in the top-level directory after getting the sources from the CVS server. (Also, unless you run configure with the *--enable-maintainer-mode* option, the configure script will not automatically get rebuilt when the *configure.in* file is updated, so you should make sure to do that manually when you notice configure.in has changed. One symptom of this is finding things like @VARIABLE@ in your Makefile after configure or *config.status* is run.)

I'm having problems configuring PHP to work with Apache. It says it can't find *httpd.h*, but it's right where I said it is!

You need to tell the configure/setup script the location of the top-level of your Apache source tree. This means that you want to specify *--with-apache=/path/to/apache* and *not --with-apache=/path/to/apache/src*.

While configuring PHP (*./configure*), you come across an error similar to the following:

```
checking lex output file root... ./configure: lex: command not found
configure: error: cannot find output from lex; giving up
```

Be sure to read the installation instructions carefully and note that you need both flex and bison installed to compile PHP. Depending on your setup you will install bison and flex from either source or a package, such as a RPM.

When I try to start Apache, I get the following message:

```
fatal: relocation error: file /path/to/libphp4.so:
symbol ap_block_alarms: referenced symbol not found
```

This error usually comes up when one compiles the Apache core program as a DSO library for shared usage. Try to reconfigure apache, making sure to use at least the following flags:

```
--enable-shared=max --enable-rule=SHARED_CORE
```

For more information, read the top-level Apache *INSTALL* file or the Apache » DSO manual page.

When I run configure, it says that it can't find the include files or library for GD, gdbm, or some other package!

You can make the configure script looks for header files and libraries in non-standard locations by specifying additional flags to pass to the C preprocessor and linker, such as:
`CPPFLAGS=-I/path/to/include LDFLAGS=-L/path/to/library ./configure`
If you're using a csh-variant for your login shell (why?), it would be:
`env CPPFLAGS=-I/path/to/include LDFLAGS=-L/path/to/library ./configure`

When it is compiling the file *language-parser.tab.c***, it gives me errors that say** *yytname undeclared***.**

You need to update your version of Bison. You can find the latest version at » http://www.gnu.org/software/bison/bison.html.

When I run *make***, it seems to run fine but then fails when it tries to link the final application complaining that it can't find some files.**

Some old versions of make that don't correctly put the compiled versions of the files in the functions directory into that same directory. Try running *cp *.o functions* and then re-running *make* to see if that helps. If it does, you should really upgrade to a recent version of GNU make.

When linking PHP, it complains about a number of undefined references.

Take a look at the link line and make sure that all of the appropriate libraries are being included at the end. Common ones that you might have missed are '-ldl' and any libraries required for any database support you included.

If you're linking with Apache 1.2.x, did you remember to add the appropriate information to the EXTRA_LIBS line of the Configuration file and re-rerun Apache's Configure script? See the installation chapter for more information.

Some people have also reported that they had to add '-ldl' immediately following *libphp4.a* when linking with Apache.

I can't figure out how to build PHP with Apache 1.3.

This is actually quite easy. Follow these steps carefully:

- Grab the latest Apache 1.3 distribution from » http://www.apache.org/dist/httpd/.
- Ungzip and untar it somewhere, for example */usr/local/src/apache-1.3*.
- Compile PHP by first running*./configure --with-apache=/<path>/apache-1.3* (substitute <path> for the actual path to your apache-1.3 directory.
- Type *make* followed by *make install* to build PHP and copy the necessary files to the Apache distribution tree.
- Change directories into to your */<path>/apache-1.3/src* directory and edit the *Configuration* file. Add to the file: *AddModule modules/php4/libphp4.a*.
- Type:*./configure* followed by *make*.
- You should now have a PHP-enabled httpd binary!

Note: You can also use the new Apache*./configure* script. See the instructions in the *README.configure* file which is part of your Apache distribution. Also have a look at the *INSTALL* file in the PHP distribution.

I have followed all the steps to install the Apache module version on Unix, and my PHP scripts show up in my browser or I am being asked to save the file.

This means that the PHP module is not getting invoked for some reason. Three things to check before asking for further help:

- Make sure that the httpd binary you are running is the actual new httpd binary you just built. To do this, try running: */path/to/binary/httpd -l* If you don't see *mod_php4.c* listed then you are not running the right binary. Find and install the correct binary.
- Make sure you have added the correct Mime Type to one of your *Apache .conf* files. It should be: *AddType application/x-httpd-php .php* Also make sure that this AddType line is not hidden away inside a <Virtualhost> or <Directory> block which would prevent it from applying to the location of your test script.
- Finally, the default location of the Apache configuration files changed between Apache 1.2 and Apache 1.3. You should check to make sure that the configuration file you are adding the AddType line to is actually being read. You can put an obvious syntax error into your *httpd.conf* file or some other obvious change that will tell you if the file is being read correctly.

It says to use: *--activate-module=src/modules/php4/libphp4.a***, but that file doesn't exist, so I changed it to** *--activate-module=src/modules/php4/libmodphp4.a* **and it doesn't work!? What's going on?**

Note that the *libphp4.a* file is not supposed to exist. The apache process will create it!

When I try to build Apache with PHP as a static module using *--activate-module=src/modules/php4/libphp4.a* **it tells me that my compiler is not ANSI compliant.**

This is a misleading error message from Apache that has been fixed in more recent versions.

When I try to build PHP using *--with-apxs* **I get strange error messages.**

There are three things to check here. First, for some reason when Apache builds the

apxs Perl script, it sometimes ends up getting built without the proper compiler and flags variables. Find your apxs script (try the command *which apxs*), it's sometimes found in */usr/local/apache/bin/apxs* or */usr/sbin/apxs*. Open it and check for lines similar to these:

```
my $CFG_CFLAGS_SHLIB  = ' ';          # substituted via Makefile.tmpl
my $CFG_LD_SHLIB      = ' ';          # substituted via Makefile.tmpl
my $CFG_LDFLAGS_SHLIB = ' ';          # substituted via Makefile.tmpl
```

If this is what you see, you have found your problem. They may contain just spaces or other incorrect values, such as 'q()'. Change these lines to say:

```
my $CFG_CFLAGS_SHLIB  = '-fpic -DSHARED_MODULE'; # substituted via
Makefile.tmpl
my $CFG_LD_SHLIB      = 'gcc';                   # substituted via
Makefile.tmpl
my $CFG_LDFLAGS_SHLIB = q(-shared);              # substituted via
Makefile.tmpl
```

The second possible problem should only be an issue on Red Hat 6.1 and 6.2. The apxs script Red Hat ships is broken. Look for this line:

```
my $CFG_LIBEXECDIR    = 'modules';         # substituted via APACI install
```

If you see the above line, change it to this:

```
my $CFG_LIBEXECDIR    = '/usr/lib/apache'; # substituted via APACI install
```

Last, if you reconfigure/reinstall Apache, add a *make clean* to the process after *./configure* and before *make*.

During *make*, I get errors in microtime, and a lot of *RUSAGE_* stuff.

During the *make* portion of installation, if you encounter problems that look similar to this:

```
microtime.c: In function `php_if_getrusage':
microtime.c:94: storage size of `usg' isn't known
microtime.c:97: `RUSAGE_SELF' undeclared (first use in this function)
microtime.c:97: (Each undeclared identifier is reported only once
microtime.c:97: for each function it appears in.)
microtime.c:103: `RUSAGE_CHILDREN' undeclared (first use in this function)
make[3]: *** [microtime.lo] Error 1
make[3]: Leaving directory `/home/master/php-4.0.1/ext/standard'
make[2]: *** [all-recursive] Error 1
make[2]: Leaving directory `/home/master/php-4.0.1/ext/standard'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory `/home/master/php-4.0.1/ext'
make: *** [all-recursive] Error 1
```

Your system is broken. You need to fix your */usr/include* files by installing a glibc-devel package that matches your glibc. This has absolutely nothing to do with PHP. To prove this to yourself, try this simple test:

```
$ cat >test.c <<X
#include <sys/resource.h>
X
$ gcc -E test.c >/dev/null
```

If that spews out errors, you know your include files are messed up.

When compiling PHP with MySQL, configure runs fine but during *make* **I get an error similar to the following:** *ext/mysql/libmysql/my_tempnam.o(.text+0x46): In function my_tempnam': /php4/ext/mysql/libmysql/my_tempnam.c:103: the use of tempnam' is dangerous, better use mkstemp'*, **what's wrong?**

First, it's important to realize that this is a *Warning* and not a fatal error. Because this is often the last output seen during *make*, it may seem like a fatal error but it's not. Of course, if you set your compiler to die on Warnings, it will. Also keep in mind that MySQL support is enabled by default.

Note

As of PHP 4.3.2, you'll also see the following text after the build (make) completes:
 	
`Build complete.`
`(It is safe to ignore warnings about tempnam and tmpnam).`

I want to upgrade my PHP. Where can I find the*./configure* **line that was used to build my current PHP installation?**

Either you look at config.nice file, in the source tree of your current PHP installation or, if this is not available, you simply run a
`<?php phpinfo(); ?>`
script. On top of the output the*./configure* line, that was used to build this PHP installation is shown.

When building PHP with the GD library it either gives strange compile errors or segfaults on execution.

Make sure your GD library and PHP are linked against the same depending libraries (e.g. libpng).

When compiling PHP I seemingly get random errors, like it hangs. I'm using Solaris if that matters.

Using non-GNU utilities while compiling PHP may cause problems. Be sure to use GNU tools in order to be certain that compiling PHP will work. For example, on Solaris, using either the SunOS BSD-compatible or Solaris versions of *sed* will not work, but using the GNU or Sun POSIX (xpg4) versions of *sed* will work. Links: » GNU sed, » GNU flex, and » GNU bison.

Using PHP

This section gathers many common errors that you may face while writing PHP scripts.

I would like to write a generic PHP script that can handle data coming from any form. How do I know which POST method variables are available?

PHP offers many predefined variables, like the superglobal *$_POST*. You may loop through *$_POST* as it's an associate array of all POSTed values. For example, let's simply loop through it with foreach, check for empty() values, and print them out.

```
<?php
$empty = $post = array();
foreach ($_POST as $varname => $varvalue) {
   if (empty($varvalue)) {
       $empty[$varname] = $varvalue;
   } else {
       $post[$varname] = $varvalue;
   }
}

print "<pre>";
if (empty($empty)) {
   print "None of the POSTed values are empty, posted:\n";
   var_dump($post);
} else {
   print "We have " . count($empty) . " empty values\n";
   print "Posted:\n"; var_dump($post);
   print "Empty:\n";  var_dump($empty);
   exit;
}
?>
```

Note
Superglobals: availability note Superglobal arrays such as *$_GET*, *$_POST*, and *$_SERVER*, etc. are available as of PHP 4.1.0. For more information, read the manual section on superglobals

I need to convert all single-quotes (') to a backslash followed by a single-quote (\'). How can I do this with a regular expression? I'd also like to convert " to \" and \ to \\.

The function addslashes() will do this. See also mysql_escape_string(). You may also

strip backslashes with stripslashes().

Note
directive note: magic_quotes_gpc The magic_quotes_gpc directive defaults to *on*. It essentially runs addslashes() on all GET, POST, and COOKIE data. stripslashes() may be used to remove them.

All my " turn into \" and my ' turn into \', how do I get rid of all these unwanted backslashes? How and why did they get there?

The PHP function stripslashes() will strip those backslashes from your string. Most likely the backslashes magically exist because the PHP directive magic_quotes_gpc is on.

Note
directive note: magic_quotes_gpc The magic_quotes_gpc directive defaults to *on*. It essentially runs addslashes() on all GET, POST, and COOKIE data. stripslashes() may be used to remove them.

How does the PHP directive register_globals affect me?

First, an explanation about what this ini setting does. Let's say the following URL is used: *http://example.com/foo.php?animal=cat* and in *foo.php* we might have the following PHP code:

```
<?php
// Using $_GET here is preferred
echo $_GET['animal'];

// For $animal to exist, register_globals must be on
// DO NOT DO THIS
echo $animal;

// This applies to all variables, so $_SERVER too
echo $_SERVER['PHP_SELF'];
```

```
// Again, for $PHP_SELF to exist, register_globals must be on
// DO NOT DO THIS
echo $PHP_SELF;
?>
```

The code above demonstrates how register_globals creates a lot of variables. For years this type of coding has been frowned upon, and for years it's been disabled by default. Note that PHP 6 removes this deprecated feature. So although most web hosts disable register_globals, there are still outdated articles, tutorials, and books that require it to be on. Plan accordingly.

See also the following resources for additional information:

- The register_globals directive
- The security chapter about register globals
- Handling external variables
- Use superglobals instead

Note
In the example above, we used an URL that contained a QUERY_STRING. Passing information like this is done through a GET HTTP Request, so this is why the superglobal *$_GET* was used.

When I do the following, the output is printed in the wrong order:

```
<?php
function myfunc($argument)
{
   echo $argument + 10;
}
$variable = 10;
echo "myfunc($variable) = " . myfunc($variable);
?>
```

what's going on?

To be able to use the results of your function in an expression (such as concatenating it with other strings in the example above), you need to **return()** the value, not echo() it.

Hey, what happened to my newlines?

```
<pre>
<?php echo "This should be the first line."; ?>
<?php echo "This should show up after the new line above."; ?>
```

```
</pre>
```

In PHP, the ending for a block of code is either "?>" or "?>\n" (where \n means a newline). So in the example above, the echoed sentences will be on one line, because PHP omits the newlines after the block ending. This means that you need to insert an extra newline after each block of PHP code to make it print out one newline.

Why does PHP do this? Because when formatting normal HTML, this usually makes your life easier because you don't want that newline, but you'd have to create extremely long lines or otherwise make the raw page source unreadable to achieve that effect.

I get the message 'Warning: Cannot send session cookie - headers already sent...' or 'Cannot add header information - headers already sent...'.

The functions header(), setcookie(), and the session functions need to add headers to the output stream but headers can only be sent before all other content. There can be no output before using these functions, output such as HTML. The function headers_sent() will check if your script has already sent headers and see also the Output Control functions.

I need to access information in the request header directly. How can I do this?

The getallheaders() function will do this if you are running PHP as an Apache module. So, the following bit of code will show you all the request headers:

```
<?php
$headers = getallheaders();
foreach ($headers as $name => $content) {
   echo "headers[$name] = $content<br />\n";
}
?>
```

See also apache_lookup_uri(), apache_response_headers(), and fsockopen()

When I try to use authentication with IIS I get 'No Input file specified'.

The security model of IIS is at fault here. This is a problem common to all CGI programs running under IIS. A workaround is to create a plain HTML file (not parsed

by PHP) as the entry page into an authenticated directory. Then use a META tag to redirect to the PHP page, or have a link to the PHP page. PHP will then recognize the authentication correctly. With the ISAPI module, this is not a problem. This should not effect other NT web servers. For more information, see: » http://support.microsoft.com/kb/q160422/ and the manual section on HTTP Authentication.

Windows: I can't access files shared on another computer using IIS

You have to change the *Go to Internet Information Services*. Locate your PHP file and go to its properties. Go to the *File Security* tab, *Edit -< Anonymous access and authentication control*.

You can fix the problem either by unticking *Anonymous Access* and leaving *Integrated Window Authentication* ticked, or, by ticking *Anonymous Access* and editing the user as he may not have the access right.

My PHP script works on IE and Lynx, but on Netscape some of my output is missing. When I do a "View Source" I see the content in IE but not in Netscape.

Netscape is more strict regarding HTML tags (such as tables) then IE. Running your HTML output through a HTML validator, such as » validator.w3.org, might be helpful. For example, a missing </table> might cause this.

Also, both IE and Lynx ignore any NULs (*\0*) in the HTML stream, Netscape does not. The best way to check for this is to compile the command line version of PHP (also known as the CGI version) and run your script from the command line. In *nix, pipe it through *od -c* and look for any *\0* characters. If you are on Windows you need to find an editor or some other program that lets you look at binary files. When Netscape sees a NUL in a file it will typically not output anything else on that line whereas both IE and Lynx will.

How am I supposed to mix XML and PHP? It complains about my <?xml tags!

In order to embed <?xml straight into your PHP code, you'll have to turn off short tags by having the PHP directive short_open_tags set to *0*. You cannot set this directive with ini_set(). Regardless of short_open_tags being on or off, you can do something like: *<?php echo '<?xml'; ?>*. The default for this directive is on.

How can I use PHP with FrontPage or some other HTML editor that insists on moving my code around?

One of the easiest things to do is to enable using ASP tags in your PHP code. This allows you to use the ASP-style <% and %> code delimiters. Some of the popular HTML editors handle those more intelligently (for now). To enable the ASP-style tags, you need to set the asp_tags *php.ini* variable, or use the appropriate Apache directive.

Where can I find a complete list of variables are available to me in PHP?

Read the manual page on predefined variables as it includes a partial list of predefined variables available to your script. A complete list of available variables (and much more information) can be seen by calling the phpinfo() function. Be sure to read the manual section on variables from outside of PHP as it describes common scenarios for external variables, like from a HTML form, a Cookie, and the URL.

Note
register_globals: important note As of PHP 4.2.0, the default value for the PHP directive register_globals is *off*, and it was completely removed as of PHP 6.0.0. The PHP community discourages developers from relying on this directive, and encourages the use of other means, such as the superglobals.

How can I generate PDF files without using the non-free and commercial libraries like PDFLib ? I'd like something that's free and doesn't require external PDF libraries.

There are a few alternatives written in PHP such as » http://www.ros.co.nz/pdf/, » http://www.fpdf.org/, » http://www.gnuvox.com/pdf4php/, and » http://www.potentialtech.com/ppl.php. There is also the » Panda module.

I'm trying to access one of the standard CGI variables (such as $DOCUMENT_ROOT or $HTTP_REFERER) in a user-defined function, and it can't seem to find it. What's wrong?

It's important to realize that the PHP directive register_globals also affects server and environment variables. When register_globals = off (the default is off since PHP 4.2.0), *$DOCUMENT_ROOT* will not exist. Instead, use *$_SERVER['DOCUMENT_ROOT']*. If register_globals = on then the variables *$DOCUMENT_ROOT* and *$GLOBALS['DOCUMENT_ROOT']* will also exist.

If you're sure register_globals = on and wonder why *$DOCUMENT_ROOT* isn't available inside functions, it's because these are like any other variables and would require *global $DOCUMENT_ROOT* inside the function. See also the manual page on variable scope. It's preferred to code with register_globals = off.

Note
Superglobals: availability note Superglobal arrays such as *$_GET*, *$_POST*, and *$_SERVER*, etc. are available as of PHP 4.1.0. For more information, read the manual section on superglobals

A few PHP directives may also take on shorthand byte values, as opposed to only integer byte values. What are all the available shorthand byte options? And can I use these outside of *php.ini* ?

The available options are K (for Kilobytes), M (for Megabytes) and G (for Gigabytes; available since PHP 5.1.0), these are case insensitive. Anything else assumes bytes. *1M* equals one Megabyte or *1048576* bytes. *1K* equals one Kilobyte or *1024* bytes. You may not use these shorthand notations outside of *php.ini*, instead use an integer value of bytes. See the ini_get() documentation for an example on how to convert these values.

PHP and HTML

PHP and HTML interact a lot: PHP can generate HTML, and HTML can pass information to PHP. Before reading these faqs, it's important you learn how to retrieve variables from external sources. The manual page on this topic includes many examples as well. Pay close attention to what *register_globals* means to you too.

What encoding/decoding do I need when I pass a value through a form/URL?

There are several stages for which encoding is important. Assuming that you have a string *$data*, which contains the string you want to pass on in a non-encoded way, these are the relevant stages:

- HTML interpretation. In order to specify a random string, you *must* include it in double quotes, and htmlspecialchars() the whole value.
- URL: A URL consists of several parts. If you want your data to be interpreted as one item, you *must* encode it with urlencode().

Example #1 - A hidden HTML form element

```
<?php
   echo "<input type='hidden' value='" . htmlspecialchars($data) . "'
/>\n";
?>
```

Note

It is wrong to urlencode() *$data*, because it's the browsers responsibility to urlencode() the data. All popular browsers do that correctly. Note that this will happen regardless of the method (i.e., GET or POST). You'll only notice this in case of GET request though, because POST requests are usually hidden.

Example #2 - Data to be edited by the user

```
<?php
   echo "<textarea name='mydata'>\n";
   echo htmlspecialchars($data)."\n";
   echo "</textarea>";
?>
```

Note

The data is shown in the browser as intended, because the browser will interpret the HTML escaped symbols.

Upon submitting, either via GET or POST, the data will be urlencoded by the browser for transferring, and directly urldecoded by PHP. So in the end, you don't need to do any urlencoding/urldecoding yourself, everything is handled automagically.

Example #3 - In a URL

```
<?php
   echo "<a href='" . htmlspecialchars("/nextpage.php?stage=23&data=" .
       urlencode($data)) . "'>\n";
?>
```

Note

In fact you are faking a HTML GET request, therefore it's necessary to manually urlencode() the data.

Note

You need to htmlspecialchars() the whole URL, because the URL occurs as value of an HTML-attribute. In this case, the browser will first un- htmlspecialchars() the value, and then pass the URL on. PHP will understand the URL correctly, because you **urlencoded()** the data.

You'll notice that the *&*in the URL is replaced by *&*. Although most browsers will recover if you forget this, this isn't always possible. So even if your URL is not dynamic, you *need* to htmlspecialchars() the URL.

I'm trying to use an <input type="image"> tag, but the *$foo.x* and *$foo.y* variables aren't available. *$_GET['foo.x']* isn't existing either. Where are they?

When submitting a form, it is possible to use an image instead of the standard submit button with a tag like:

```
<input type="image" src="image.gif" name="foo" />
```

When the user clicks somewhere on the image, the accompanying form will be

transmitted to the server with two additional variables: *foo.x* and *foo.y*.

Because *foo.x* and *foo.y* would make invalid variable names in PHP, they are automagically converted to *foo_x* and *foo_y*. That is, the periods are replaced with underscores. So, you'd access these variables like any other described within the section on retrieving variables from external sources. For example, *$_GET['foo_x']*.

Note
Spaces in request variable names are converted to underscores.

How do I create arrays in a HTML <form>?

To get your <form> result sent as an array to your PHP script you name the <input>, <select> or <textarea> elements like this:

```
<input name="MyArray[]" />
<input name="MyArray[]" />
<input name="MyArray[]" />
<input name="MyArray[]" />
```

Notice the square brackets after the variable name, that's what makes it an array. You can group the elements into different arrays by assigning the same name to different elements:

```
<input name="MyArray[]" />
<input name="MyArray[]" />
<input name="MyOtherArray[]" />
<input name="MyOtherArray[]" />
```

This produces two arrays, MyArray and MyOtherArray, that gets sent to the PHP script. It's also possible to assign specific keys to your arrays:

```
<input name="AnotherArray[]" />
<input name="AnotherArray[]" />
<input name="AnotherArray[email]" />
<input name="AnotherArray[phone]" />
```

The AnotherArray array will now contain the keys 0, 1, email and phone.

Note
Specifying an arrays key is optional in HTML. If you do not specify the keys, the array gets filled in the order the elements appear in the form. Our first example will contain keys 0, 1, 2 and 3.

See also Array Functions and Variables From External Sources.

How do I get all the results from a select multiple HTML tag?

The select multiple tag in an HTML construct allows users to select multiple items from a list. These items are then passed to the action handler for the form. The problem is that they are all passed with the same widget name. I.e.

```
<select name="var" multiple="yes">
```

Each selected option will arrive at the action handler as:

```
var=option1
var=option2
var=option3
```

Each option will overwrite the contents of the previous *$var* variable. The solution is to use PHP's "array from form element" feature. The following should be used:

```
<select name="var[]" multiple="yes">
```

This tells PHP to treat *$var* as an array and each assignment of a value to var[] adds an item to the array. The first item becomes *$var[0]*, the next *$var[1]*, etc. The count() function can be used to determine how many options were selected, and the sort() function can be used to sort the option array if necessary.

Note that if you are using JavaScript the *[]* on the element name might cause you problems when you try to refer to the element by name. Use it's numerical form element ID instead, or enclose the variable name in single quotes and use that as the index to the elements array, for example:

```
variable = documents.forms[0].elements['var[]'];
```

How can I pass a variable from Javascript to PHP?

Since Javascript is (usually) a client-side technology, and PHP is (usually) a server-side technology, and since HTTP is a "stateless" protocol, the two languages cannot directly share variables.

It is, however, possible to pass variables between the two. One way of accomplishing this is to generate Javascript code with PHP, and have the browser refresh itself, passing specific variables back to the PHP script. The example below shows precisely how to do this -- it allows PHP code to capture screen height and width, something that is normally only possible on the client side.

```
<?php
if (isset($_GET['width']) AND isset($_GET['height'])) {
 // output the geometry variables
 echo "Screen width is: ". $_GET['width'] ."<br />\n";
 echo "Screen height is: ". $_GET['height'] ."<br />\n";
} else {
 // pass the geometry variables
 // (preserve the original query string
```

```
 //   -- post variables will need to handled differently)

 echo "<script language='javascript'>\n";
 echo "
location.href=\"${_SERVER['SCRIPT_NAME']}?${_SERVER['QUERY_STRING']}"
           . "&width=\" + screen.width + \"&height=\" + screen.height;\n";
 echo "</script>\n";
 exit();
}
?>
```

PHP and COM

PHP can be used to access COM and DCOM objects on Win32 platforms.

I have built a DLL to calculate something. Is there any way to run this DLL under PHP ?

If this is a simple DLL there is no way yet to run it from PHP. If the DLL contains a COM server you may be able to access it if it implements the IDispatch interface.

What does 'Unsupported variant type: xxxx (0xxxxx)' mean ?

There are dozens of VARIANT types and combinations of them. Most of them are already supported but a few still have to be implemented. Arrays are not completely supported. Only single dimensional indexed only arrays can be passed between PHP and COM. If you find other types that aren't supported, please report them as a bug (if not already reported) and provide as much information as available.

Is it possible manipulate visual objects in PHP ?

Generally it is, but as PHP is mostly used as a web scripting language it runs in the web servers context, thus visual objects will never appear on the servers desktop. If you use PHP for application scripting e.g. in conjunction with PHP-GTK there is no limitation in accessing and manipulating visual objects through COM.

Can I store a COM object in a session ?

No, you can't. COM instances are treated as resources and therefore they are only available in a single script's context.

How can I trap COM errors ?

In PHP 5, the COM extension throws *com_exception* exceptions, which you can catch and then inspect the *code* member to determine what to do next.

In PHP 4 it's not possible to trap COM errors beside the ways provided by PHP itself (@, track_errors, ..).

Can I generate DLL files from PHP scripts like i can in Perl ?

No, unfortunately there is no such tool available for PHP.

What does 'Unable to obtain IDispatch interface for CLSID {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}' mean ?

This error can have multiple reasons:

- the CLSID is wrong
- the requested DLL is missing
- the requested component doesn't implement the IDispatch interface

How can I run COM object from remote server ?

Exactly like you run local objects. You only have to pass the IP of the remote machine as second parameter to the COM constructor.

Make sure that you have set = **TRUE** in your *php.ini*.

I get 'DCOM is disabled in C:\path...\scriptname.php on line 6', what can I do ?

Edit your *php.ini* and set = **TRUE**.

Is it possible to load/manipulate an ActiveX object in a page with PHP ?

This has nothing to do with PHP. ActiveX objects are loaded on client side if they are requested by the HTML document. There is no relation to the PHP script and therefore there is no direct server side interaction possible.

Is it possible to get a running instance of a component ?

This is possible with the help of monikers. If you want to get multiple references to the same word instance you can create that instance like shown:

```
<?php
$word = new COM("C:\docs\word.doc");
?>
```

This will create a new instance if there is no running instance available or it will return a handle to the running instance, if available.

Is there a way to handle an event sent from COM object ?

You can define an event sink and bind it using com_event_sink(). You can use com_print_typeinfo() to have PHP generate a skeleton for the event sink class.

I'm having problems when trying to invoke a method of a COM object which exposes more than one interface. What can I do ?

The answer is as simple as unsatisfying. I don't know exactly but i think you can do nothing. If someone has specific information about this, please let » me know :)

So PHP works with COM, how about COM+ ?

COM+ extends COM by a framework for managing components through MTS and MSMQ but there is nothing special that PHP has to support to use such components.

If PHP can manipulate COM objects, can we imagine to use MTS to manage components resources, in conjunction with PHP ?

PHP itself doesn't handle transactions yet. Thus if an error occurs no rollback is initiated. If you use components that support transactions you will have to implement the transaction management yourself.

PHP and other languages

PHP is the best language for web programing, but what about other languages?

PHP vs. ASP?

ASP is not really a language in itself, it's an acronym for Active Server Pages, the actual language used to program ASP with is Visual Basic Script or JScript. The biggest drawback of ASP is that it's a proprietary system that is natively used only on Microsoft Internet Information Server (IIS). This limits it's availability to Win32 based servers. There are a couple of projects in the works that allows ASP to run in other environments and webservers: » InstantASP from » Halcyon (commercial), Chili!Soft ASP from » Chili!Soft (commercial). ASP is said to be a slower and more cumbersome language than PHP, less stable as well. Some of the pros of ASP is that since it primarily uses VBScript it's relatively easy to pick up the language if you're already know how to program in Visual Basic. ASP support is also enabled by default in the IIS server making it easy to get up and running. The components built in ASP are really limited, so if you need to use "advanced" features like interacting with FTP servers, you need to buy additional components.

Is there an ASP to PHP converter?

Yes, the server-side » asp2php is the one most often referred to as well as » this client-side option.

PHP vs. Cold Fusion?

PHP is commonly said to be faster and more efficient for complex programming tasks and trying out new ideas. PHP is generally referred to as more stable and less resource intensive as well. Cold Fusion has better error handling, database abstraction and date parsing although database abstraction is addressed in PHP 4. Another thing that is listed as one of Cold Fusion's strengths is its excellent search engine, but it has been mentioned that a search engine is not something that should be included in a web scripting language. PHP runs on almost every platform there is; Cold Fusion is only available on Win32, Solaris, Linux and HP/UX. Cold Fusion has a good IDE and is generally easier to get started with, whereas PHP initially requires more programming knowledge. Cold Fusion is designed with non-programmers in mind, while PHP is focused on programmers.

A great summary by Michael J Sheldon on this topic has been posted to the PHP mailing list. A copy can be found at » http://marc.theaimsgroup.com/?l=php-general&m=95602167412542&w=1.

PHP vs. Perl?

The biggest advantage of PHP over Perl is that PHP was designed for scripting for the web where Perl was designed to do a lot more and can because of this get very complicated. The flexibility / complexity of Perl makes it easier to write code that another author / coder has a hard time reading. PHP has a less confusing and stricter format without losing flexibility. PHP is easier to integrate into existing HTML than Perl. PHP has pretty much all the 'good' functionality of Perl: constructs, syntax and so on, without making it as complicated as Perl can be. Perl is a very tried and true language, it's been around since the late eighties, but PHP is maturing very quickly.

Migrating from PHP 4 to PHP 5

This faq section will help you migrate from PHP 4 to PHP 5.

Migrating from PHP 4 to PHP 5

Although PHP 5 offers many new features, it's designed to be as compatible with earlier versions of PHP as possible with little functionality being broken in the process.

Be sure to read the appropriate PHP 5 migration appendix of this manual as it contains even more information on the topic of migrating to PHP 5.

Does MySQL work in PHP 5? It seemed to have disappeared.

MySQL is supported with the only change being that MySQL support is no longer enabled by *default* in PHP 5. This essentially means that PHP doesn't include the *--with-mysql* option in the configure line so that you must now manually do this when compiling PHP. Windows users will edit *php.ini* and enable the *php_mysql.dll* DLL as in PHP 4 no such DLL existed, it was simply built into your Windows PHP binaries.

Also, the MySQL client libraries are no longer bundled with PHP. More details on this topic are covered in the following FAQ and be sure to read the MySQL section for details on installing MySQL. An example configure line would be *--with-mysql=/usr* while Windows users will need the *libmySQL.dll* available to the system.

I hear PHP 5 has an entirely new OOP model, will my existing OOP code work? Where do I find information on these new OOP features?

The main change in PHP 5 is to the OOP model as PHP 5 now uses the *Zend Engine 2.0*. The zend.ze1_compatibility_mode directive enables compatability with the Zend Engine 1.0 (PHP 4).

The new OOP model is documented in the OOP language reference and OOP migration appendix sections.

So besides the new OOP model, what else has changed in PHP 5? Also, is there a

PHP 5 specific version of the PHP manual?

Few other changes exist, see the migration 5 appendix for details. There won't be a PHP 5 specific version of the manual as the bulk of PHP remains the same.

Miscellaneous Questions

There can be some questions we can't put into other categories. Here you can find them.

How can I handle the bz2 compressed manuals on Windows?

If you don't have an archiver-tool to handle bz2 files » download the command line tool from Redhat (please find further information below).

If you would not like to use a command line tool, you can try free tools like » Stuffit Expander, » UltimateZip, » 7-Zip, or » Quick Zip. If you have tools like » WinRAR or » Power Archiver, you can easily decompress the bz2 files with it. If you use Total Commander (formerly Windows Commander), a bz2 plugin for that program is available freely from the » Total Commander site.

The bzip2 command line tool from Redhat:

Win2k Sp2 users grab the latest version 1.0.2, all other Windows user should grab version 1.00. After downloading rename the executable to bzip2.exe. For convenience put it into a directory in your path, e.g. C:\Windows where C represents your Windows installation drive.

Note: lang stands for your language and x for the desired format, e.g.: pdf. To uncompress the php_manual_lang.x.bz2 follow these simple instructions:

- open a command prompt window
- cd to the folder where you stored the downloaded php_manual_lang.x.bz2
- invoke bzip2 -d php_manual_lang.x.bz2, extracting php_manual_lang.x in the same folder

In case you downloaded the php_manual_lang.tar.bz2 with many html-files in it, the procedure is the same. The only difference is that you got a file php_manual_lang.tar. The tar format is known to be treated with most common archivers on Windows like e.g. » WinZip.

What does & beside argument mean in function declaration of e.g. asort() ?

It means that the argument is passed by reference and the function will likely modify it corresponding to the documentation. You can pass only variables this way and you don't need to pass them with & in function call (it's even deprecated).

How do I deal with *register_globals* ?

For information about the security implications of *register_globals*, read the security chapter on Using register_globals.

It's preferred to use superglobals, rather than relying upon *register_globals* being on.

If you are on a shared host with *register_globals* turned off and need to use some legacy applications, which require this option to be turned on, or you are on some hosting server, where this feature is turned on, but you would like to eliminate security risks, you might need to emulate the opposite setting with PHP. It is always a good idea to first ask if it would be possible to change the option somehow in PHP's configuration, but if it is not possible, then you can use these compatibility snippets.

Example #4 - Emulating Register Globals

This will emulate register_globals On. If you altered your variables_order directive, consider changing the *$superglobals* accordingly.

```
<?php
// Emulate register_globals on
if (!ini_get('register_globals')) {
   $superglobals = array($_SERVER, $_ENV,
       $_FILES, $_COOKIE, $_POST, $_GET);
   if (isset($_SESSION)) {
       array_unshift($superglobals, $_SESSION);
   }
   foreach ($superglobals as $superglobal) {
       extract($superglobal, EXTR_SKIP);
   }
}
?>
```

This will emulate register_globals Off. Keep in mind, that this code should be called at the very beginning of your script, or after session_start() if you use it to start your session.

```
<?php
// Emulate register_globals off
function unregister_GLOBALS()
{
   if (!ini_get('register_globals')) {
       return;
   }

   // Might want to change this perhaps to a nicer error
   if (isset($_REQUEST['GLOBALS']) || isset($_FILES['GLOBALS'])) {
       die('GLOBALS overwrite attempt detected');
   }

   // Variables that shouldn't be unset
   $noUnset = array('GLOBALS',  '_GET',
                    '_POST',    '_COOKIE',
```

```
                    '_REQUEST', '_SERVER',
                    '_ENV',     '_FILES');

    $input = array_merge($_GET,    $_POST,
                         $_COOKIE, $_SERVER,
                         $_ENV,    $_FILES,
                         isset($_SESSION) && is_array($_SESSION) ?
$_SESSION : array());

    foreach ($input as $k => $v) {
        if (!in_array($k, $noUnset) && isset($GLOBALS[$k])) {
            unset($GLOBALS[$k]);
        }
    }
}

unregister_GLOBALS();

?>
```