# Sybase−PHP−Apache mini−HOWTO


## Tyson Lloyd Thwaites

tyson.lloydthwaites@ite.com.au



**Revision History**

| | | |
|---|---|---|
| Revision 1.3 | 2001−07−11 | Revised by: TLT |
| Fixed example PHP script. | | |
| Revision 1.2 | 2001−06−13 | Revised by: TLT |
| Shane Gelven pointed out an error in the init script paths. | | |
| Revision 1.1 | 2001−05−29 | Revised by: TLT |
| Updated environment configuration section, changed HOWTO name. | | |



This HOWTO explains how to set up a Linux machine to run an Apache web server using PHP to access a Sybase−ASE database.

# Table of Contents

# 1. Copyright and License

# 2. Install Linux

This howto assumes you have already installed a working Linux system. This howto is based on a Redhat 6.2 system. I would appreciate any feedback on setting this up on other distros.

The first thing you will have to do is uninstall the `apache` and `mod_php` rpms. Try the following:

```
# rpm -e apache
# rpm -e httpd
# rpm -e mod_php
# rpm -e php
```

That should get rid of them. Just to be sure, run this:

```
# rpm -qa | less
```

and check that there is nothing to do with Apache or PHP left on you system.

# 3. Install Sybase

Download the Sybase−ASE files from linux.sybase.com. Sybase−ASE 11.0.3.3 is free for production. This howto refers to that version.

First, install the Sybase RPMS as root:

```
    # rpm −ivh sybase−ase−11_0_3_3−6_i386.rpm
    # rpm −ivh sybase−doc−11_0_3_3−6_i386.rpm
    # rpm −ivh sybase−ocsd−10_0_4−6_i386.rpm
```

For information on how to configure Sybase, read the instructions in the file `/opt/sybase/doc/howto/howto−ase−quickstart.html`. This document is very good, so we shouldn't need any more information here.

> **Hint:** After you have installed the sample database and run a select against it, you can stop following the instructions. The rest of the howto goes on to uninstall the sample database, but we want to keep it for testing later.
>
> **NOTE:** Some people have reported problems with the way the Sybase init scripts try to guess the Sybase directory. If you see the following error:
>
> ```
>     export: sybase−ase−11.0.3.3: not a legal variable name
> ```
>
> when trying to run sybinstall.sh, then you need to edit the sybinstall.sh. Open the script and find the following lines (around line 104):
>
> ```
> # export SYBASE=/opt/sybase
> export SYBASE=`rpm −q −−queryformat \
>                 '%{installprefix}\n' sybase−ase−11.0.3.3`
> ```
>
> Change these lines so they look like this:
>
> ```
> export SYBASE=/opt/sybase
> #export SYBASE=`rpm −q −−queryformat \
> #               '%{installprefix}\n' sybase−ase−11.0.3.3`
> ```
>
> This hardcodes the SYBASE path to /opt/sybase. If you did install Sybase somewhere non−standard and you get this error, you will have to change this path to point to your Sybase directory.

# 4. Configure Environment

After installing Sybase, you need to setup the Linux environment in order to allow clients to connect to the dataserver.

The first thing we need to do is to set certain environment variables in `/etc/profile`. Open `/etc/profile` as super user and insert the following lines:

```
    export SYBASE=/opt/sybase
    export SYBPLATFORM=linux
    export LD_LIBRARY_PATH=$SYBASE/lib
    export LC_ALL=default
    export PATH="$SYBASE/bin:$PATH"
    export DSQUERY=SYBASE
```

Next you need to change to permissions of the Sybase directory. This fixes some problems where PHP is unable to connect to the Sybase server. Run the following command as super user:

```
chown -R sybase:sybase /opt/sybase
```

To test whether your Sybase environment works, type the following:

```
    # su - nobody
    $ isql -Usa -P
    1> sp_helpdb
    2> go
```

You should see a list of databases. To test whether you can use the sample datebase, now type:

```
    1> use pubs2
    2> go
    1> select * from titles
    2> go
```

If you a list, you have sucessfully set up Sybase.

> **NOTE:** You may need to do the following in order for PHP to be able to access Sybase. Edit the file /opt/sybase/install/rc.sybase and remove the line that says:
>
> ```
>     unset LANG; unset LC_ALL; \
> ```
>
> I don't know why that line is there, but it can stuff things up.
>
> **NOTE:** Some people have reported problems with the way the Sybase init scripts try to guess the Sybase directory. If you see the following error:
>
> ```
>     errorerwerqwerqwerwerwer
> ```
>
> when trying to run sybinstall.sh or rc.sybase, then you need to change

You may also want to add sybase to the list of services to run at boot time. To do this execute the following commands:

```
# ln -s /opt/sybase/install/rc.sybase /etc/rc.d/init.d/sybase
# ln -s /etc/rc.d/init.d/sybase /etc/rc.d/rc3.d/S15sybase
# ln -s /etc/rc.d/init.d/sybase /etc/rc.d/rc3.d/K15sybase
```

Now Sybase will start whenever you boot into runlevel 3.

> **NOTE:** The note for

## 4.1. Preconfigure Apache

Download and unpack the latest Apache distribution. In the Apache directory issue the following command:

```
# ./configure --prefix=/usr/local/apache
```

I think the 'prefix' value specifies where you want to store your html files; I just set it to the same as the apache install directory, where Apache will put its files when you run '**make install**'.

# 5. Install PHP

Download the latest PHP distribution and unpack it.

Change to the `php` directory and run the **configure** script. You need to pass certains options to the script. To enable PHP access Sybase, you need **−−with−sybase−ct**. You will also need to specify the web server interface to use; in our case, this is Apache, so we use **−−with−apache** To build a CGI version of PHP, (which is also very useful as a shell scripting tool), just leave out the −−with−apache.

Here is a complete configure script with the above two options, as well as a number of other options that are useful.

```
    # ./configure --with-apache=../apache_1.3.19
      --with-sybase-ct=/opt/sybase --enable-bcmath --enable-calendar
      --enable-ctype --enable-exif --enable-ftp --enable-gd-imgstrttf
      --with-gd --enable-trans-sid --enable-shmop --enable-sockets
      --enable-sysvsem --enable-sysvshm --enable-wddx
```

You will need to change to apache path to wherever you unpacked your apache distribution.

If the configure scripts completes successfully, you can then proceed:

```
    # make && make install
```

PHP should compile without any hitches. After it has installed, you need to install the `php.ini` file:

```
    # cp php.ini-optimized /usr/local/lib/php.ini
```

Now we need to change a few things in `/usr/local/lib/php.ini`. Set the following parameters to the ones specified below:

```
    magic_quotes_sybase = on
    sybct.min_server_severity = 11
    sybct.min_client_severity = 11
```

I also like to set the following:

```
    register_globals = on
    include_path=.:/usr/local/lib/php
```

You have now completed the PHP installation.

# 6. Install Apache

Now **cd** back into the apache unpack directory. Issue the following commmands:

```
# ./configure --activate-module=src/modules/php4/libphp4.a
# make && make install
```

This should proceed without any problems. When its done, edit the file `/usr/local/apache/conf/httpd.conf` and uncomment the following line:

```
AddType application/x-httpd-php .php
```

This tells apache to allow PHP to process files with a `.php` extension. You may also want to add some other extensions to the end of this line, eg:

```
AddType application/x-httpd-php .php .php3 .html .htm
```

You now have all the software you need and are ready to test your system.

# 7. Test the System

First, make sure Apache and Sybase are started:

```
    # /usr/local/apache/bin/apachectl start
    # /opt/sybase/install/rc.sybase start
```

Now go to the directory `/usr/local/apache/htdocs`. Create a file called `test.php` and insert the following contents:

```
        <?php
                $con = sybase_connect("SYBASE", "sa", "");
                sybase_select_db("pubs2");
                $qry = sybase_query("select * from titles", $con);

                echo sybase_result($qry, 1, 1);

                sybase_close($con);
        ?>
```

Save the file and then access the following URL in your browser:

```
    http://localhost/test.php
```

If you see some text, then congratulations! You have a complete Linux/Apache/PHP/Sybase−ASE installation. If not...well, there you go. Make sure you actually have installed the pubs2 database. If you are not sure, try running:

```
        # isql −Usa −P < /opt/sybase/scripts/installpubs2
```

Email me if you have any problems, corrections (please!) or additions, but please do not email me unless you have tried everything in this howto! Thanks.

# 8. Post Install Tasks

After you have a working system, first of all, breath a deep sigh of satisfaction. (I also suggest eating something.) After this, there are a few post−install tasks you should perform.

The FIRST thing you should do is change the password for the 'sa' user. By default this password is blank. To set a password for sa, type the following in isql:

```
1> sp_password null, 'new_pass123', 'sa'
2> go
```

This will set the password. You will now need to pass this as a parameter to sybase_connect. You will also need to specify it when starting isql, which you can do by using the '−P' switch.

> **NOTE:** Once you have ste a password, you cannot unset it. (Sybase requires a minumum of six characters in a password, so '' won't be accepted.)

You also may wish to install the Windows client tools. These can be found on a CD if you bought a cope of ASE from Sybase. Otherwise, if you are using the free 11.0.3.3 version, you can download a demo of ASE 12.0 for NT, and only install the 'ASE Plugin for Sybase Central', 'ASE ODBC Driver' and 'Sybase Central' components.

> **:** I am not sure of the licensing issues involved in using the client tools if you haven't purchased ASE. Although I don't think it would be a problem, you should contact Sybase directly if you want to be sure.

I am open to more hints to put in this section.