##  [KB 5460][Info] SiteBuilder fails to publish because of missing SQLite 2.x support

**Resolution**

The "sqlite2" libraries should be installed on the server with appropriate SQLite 2.x UTF-8 encoding support.
The SQLite libraries package can be installed on the server directly using a system-automated update program. For example, you can install it with the help of "yum":

**# yum install sqlite2**

 Alternatively, it can be found on one of the RPM resources, such as <http://rpm.pbone.net> or <http://rpmfind.net/>.

As for SQLite 2.x UTF-8 encoding support for PHP, the appropriate **php-sqlite** package which includes the necessary **sqlite.so** extension should be installed on the server. You can try to install it with help of "yum" or "apt-get" (on Debian and Ubuntu servers). For example, to install it with php5, use one of the following commands:

**# yum install php-sqlite**
or
**# apt-get install php5-sqlite** (on Ubuntu and Debian servers)

If no suitable packages are found, the extension can be also compiled from the SQLite sources. You can find instructions on how to compile sqlite.so with UTF-8 encoding support below.

**Note:**the following packages should be installed for this operation: **tar, php, php-devel**

**These are general instructions. The appropriate actions may differ from these depending on your OS installation and configuration.**

***Step-by-step instructions are listed below:***

1. Download the source package. You can get it here: <http://pecl.php.net/package/SQLite>.

***# wget http://pecl.php.net/get/SQLite-1.0.3.tgz***
or
**# pear download sqlite**If the publication host has an x86\_64 architecture, it is suggested that you use the latest SQLite extension source files from the corresponding PHP version. Also make sure that the SQLite library has a version number of 2.8.17 or greater.

1. Unpack*SQLite-1.0.3.tgz*to the *SQLite-1.0.3* directory:

***# tar -xzf SQLite-1.0.3.tgz***

1. Move into the *SQLite-1.0.3* directory:

***# cd SQLite-1.0.3***

1. You need to have UTF-8 encoding support for the sqlite.so extension, so the following modification should be made:

edit the *SQLite-1.0.3/libsqlite/src/main.c* file and change this section:

***#ifdef SQLITE\_UTF8
const char sqlite\_encoding[] = "UTF-8";
#else
const char sqlite\_encoding[] = "iso8859";
#endif***

 to

***//#ifdef SQLITE\_UTF8
const char sqlite\_encoding[] = "UTF-8";
//#else
//const char sqlite\_encoding[] = "iso8859";
//#endif***

 Comment all strings, excluding the one with UTF-8 encoding.

1. Find the *phpize* binary file and run it. The standard file path to locate this file is */usr/bin/phpize*, but yours may be different.Make sure that you are still in the *SQLite-1.0.3* directory and run the following command:

***# /usr/bin/phpize***(Path to *phpize* may be different.)

If there is no such file, try to find it with the following commands:

***# whereis phpize***
***# locate phpize|grep bin***

1. Find the location of the *php-config* file (should be the same as for the *phpize* file). Run the following commands:

***# ./configure --with-sqlite --with-php-config=/usr/bin/php-config*** (path to php-config can be different)
2. Run "*make"*command:

***# make***

If you encounter something like this:

***"error: `BYREF\_NONE' undeclared"***

... then you will need to edit *sqlite.c.* Comment out the following line:

***/\* static unsigned char arg3\_force\_ref[] = {3, BYREF\_NONE, BYREF\_NONE, BYREF\_FORCE }; \*/***

 ... and then change these lines:

***function\_entry sqlite\_functions[] = {
PHP\_FE(sqlite\_open, arg3\_force\_ref)
PHP\_FE(sqlite\_popen, arg3\_force\_ref)***

to

***function\_entry sqlite\_functions[] = {
PHP\_FE(sqlite\_open, third\_arg\_force\_ref)
PHP\_FE(sqlite\_popen, third\_arg\_force\_ref)***

1. ***# make***
2. ***# make install***
3. **# *cp modules/sqlite.so /usr/lib/php/modules*** (path to modules folder may be different). Actually, ***"make install"*** should copy the extension to the appropriate place, so this operation might not be required.
4. Create a file in */etc/php.d/* named *sqlite.ini* or add the *sqlite.so* extension directly into the php.ini file. Write this inside the file:

***; Enable sqlite extension module
extension=sqlite.so***
5. Restart Apache service with one of the following commands:

***# /sbin/service/httpd restart***
or
***# /etc/init.d/httpd restart***

***Specific of compilation of SQLite extension for PHP 5.3:***

1. Get sqlite2 module sources using ‘svn’ command (install package ‘subversion’ if needed):

***# mkdir /root/sqlite
# cd /root/sqlite
# svn co http://svn.php.net/repository/php/php-src/branches/PHP\_5\_3/ext/sqlite/***

1. Find the phpize binary file and run it. The standard file path to locate this file is /usr/bin/phpize, but yours may be different. Make sure that you are still in the SQLite-1.0.3 directory and run the following command:

***# /usr/bin/phpize*** (Path to phpize may be different.)

If there is no such file, try to find it with the following commands:

***# whereis phpize
# locate phpize|grep bin***

1. Find the location of the php-config file (should be the same as for the phpize file). Run the following commands:

***# ./configure --with-sqlite –enable-sqlite-utf8 --with-php-config=/usr/bin/php-config*** (path to php-config can be different)

1. Run "make" command:

***# make***

1. ***# cp modules/sqlite.so /usr/lib/php/modules*** (path to modules folder may be different).
2. Create a file in /etc/php.d/ named sqlite.ini or add the sqlite.so extension directly into the php.ini file. Write this inside the file:

; Enable sqlite extension module
extension=sqlite.so

1. Restart Apache service with one of the following commands:

***# /sbin/service/httpd restart***
or
***# /etc/init.d/httpd restart***