Perl DBD::mysql Module

Perl DBD::mysql Module

For some reason, if you install Perl on OpenIndiana, you will not get the DBD::mysql module with it. Of course, this is very important if you intend to use Perl to do any MySql database operations.

This omission is nothing new to OpenIndiana. It's been the case with Solaris for quite some time. To make the problem even worse, since Perl was compiled with Solaris Studio, not gcc, the DBD::mysql module also needs to be compiled with it.

If you do a search on Google about this, you'll be amazed at how many people through the years have fought, and fought, and fought with this problem on their various Solaris systems. Fortunately, I found the solution at the **bottom** of this page: http://nxadm.wordpress.com/2009/08/13/install-dbdmysql-for-solaris-10-system-perl/ This is the remarkably simple answer to something that I struggled at least eight hours with – and many others have struggled much more than that with it.

Here are the steps:

1.) Install header-math package:

pkg install header-math

2.) You will also need to install Perl. These are the packages I installed. If you don't need apache-22, feel free to take it out.

pkg install apache-22 perl-510 runtime/perl-510/extra runtime/perl-510/module/sun-solaris library/perl-5/database library/perl-5/xml-parser

3.) You will also need MySql server running on the localhost. I'm not sure yet if you will need it after this is installed. The install procedure needs at least some of the libraries, and make test needs the server running to perform its tests. I installed the following, and it worked for me to get it installed:

pkg install database/mysql-51 database/mysql-51/library database/mysql-common library/apr-util-13/dbd-mysql

4.) Configure MySql to run. I suggest setting the following in your my.cnf file first, as I had to do this on one of our systems:

Change "thread_stack = 128K" to "thread_stack = 256K" or larger.

If you are on a 64 bit system, change MySql to run in 64 bit mode:

svccfg -s mysql:version_51 setprop mysql/enable_64bit=true
svcadm refresh mysql:version_51

Enable mysql:

svcadm enable mysql

5.) When you go to actually install the DBD::mysql Module, you will see the following, so you may as well do this up front.

PLEASE NOTE:

For 'make test' to run properly, you must ensure that the
database user '' can connect to your MySQL server
and has the proper privileges that these tests require such
as 'drop table', 'create table', 'drop procedure', 'create procedure'
as well as others.

mysql> grant all privileges on test.* to ''@'localhost' identified by 's3kr1t';

You can also optionally set the user to run 'make test' with:

perl Makefile.PL --testuser=username

6.) Download SolarisStudio from Oracle's website: http://www.oracle.com/technetwork/server-storage/solarisstudio/downloads/index-jsp-141149.html

7.) Create a directory for it. I put it in /usr/solarisstudio

mkdir /usr/solarisstudio

Put it in there. Check the md5sum (optional). Unpack it.

You should now have the following directory:

/usr/solarisstudio/SolarisStudio12.3-solaris-x86-bin

8.) In a terminal, do the following. If you're using a different architecture, or a different version of SolarisStudio, adjust accordingly.

export PATH=$PATH:/usr/mysql/bin:/usr/solarisstudio/SolarisStudio12.3-solaris-x86-bin/solarisstudio12.3/bin/

The mysql part of the PATH is so that the build process can find ‘mysql_config’ file. The solarisstudio part of the PATH is so it can find the ‘cc’ compiler.

We are assuming you already have mysql installed. If not, install it now.

9.) Do this command to make sure the build process can find the compiler.

export CC=cc

10.) Now use perl's cpan program to download and install it.

perl -MCPAN -e 'install DBD::mysql'

Notice we deliberately do not use ‘perlgcc’, since we're now in the fortunate situation that we have ‘cc’ installed.

You will see warnings during the compilation. This is normal and generally not to be alarmed about. ‘cc’ is more strict about giving warnings on poorly written source (but semantically ok) than is the case for ‘gcc’.

So – despite the many warnings – it should complete with success.

Disclaimer: I didn't take this detailed of notes when I set this up. When I build our next server, I'll go through this step by step to verify everything, and make any changes to this page as necessary. It should be pretty close, though. PS - I've added more steps now that I have gone through this a second time.

I hope it helps!