

Compiling and running DBD for Oracle 10g without installing Oracle

Source: <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2004-04/0122.html>

From: Blach Johannes (johannes.blach_at_alcatel.at)

Date: 04/13/04

To: dbi-users@perl.org

Date: Tue, 13 Apr 2004 14:17:34 +0200

Hello,

I recently had to compile the DBD modules for Oracle 10g on machines, where I wasn't able to install Oracle. To save others the pain of figuring out this on their own, here's a short article of the type I was hoping to find in the first place.

These instructions worked on Sun Solaris 9 and a fragmentary installation Slackware Linux.

Ingredients:

To compile you need:

DBD-Oracle-1.15 or other recent versions
gcc of recent enough version. I uses 3.3.3, but others should work ok too
perl, DBI etc. of recent enough version
unzip most likely already installed by the OS

Oracle 10g installation disk for your architecture (available at <http://otn.oracle.com/software/products/database/oracle10g/index.html>). Only the first disk is necessary, which is a 700MB cpio archive.

To run:

perl as used above
DBI and DBD as compiled above
Oracle 10g instant client (available at <http://otn.oracle.com/software/tech/oci/instantclient/index.html>) Only the basic package is necessary (44MB), although sqlplus is nice to have too (add 0.5 MB) to test the connections.

Step one: Extracting necessary libraries from Oracle distribution

To compile the Oracle DBD, only a very small subset of the Oracle

perl.dbi.users: Compiling and running DBD for Oracle 10g without installing Oracle

installation are necessary. These files can easily be extracted directly without the need to install Oracle. For me, this was critical, as I was very short on disk space on my machine and the Oracle installer was complaining about my version of Linux. It wasn't red-hat enough for it's taste.

Once the cpio-archive is unpacked, you'll have many files jar files in a directory tree in the directory Disk1. The Oracle installer is a Java application that works itself through all Java packages installing whatever necessary on the way. Fortunately, the files necessary to compile DBD can easy extracted with a normal unzip from these archives:

```
Disk1/stage/Components/oracle.rdbms.oci/10.1.0.2.0/1/DataFiles/include.jar
Disk1/stage/Components/oracle.rsf.rdbms_rsf/10.1.0.2.0/1/DataFiles/lib.jar
Disk1/stage/Components/oracle.rdbms/10.1.0.2.0/1/DataFiles/lib.jar
Disk1/stage/Components/oracle.rsf.clntsh_rsf/10.1.0.2.0/1/DataFiles/lib.jar
```

Just run unzip whatever.jar in your temporary ORACLE_HOME directory (later referred to as /tmp/foo). If disk-space is really tight, you could just extract with cpio those 4 files.

Step two: Compiling DBD-Oracle

After unzipping the necessary files into your /tmp/foo directory, set ORACLE_HOME temporarily to this directory with

```
export ORACLE_HOME=/tmp/foo
```

DBD-Oracle can now be compiled and installed as per instruction in the fine README-files. On Linux, you're done.

Step two-bis: Compiling DBD-Oracle under Solaris with gcc

The perl version distributed by Sun in the default installation is compiled with a sun compiler. If the only compiler you have is gcc from <http://www.sunfreeware.com>, making compatible libraries isn't easy. A better guess would be to get perl 5.8 from sunfreeware too, which is already compiled with gcc.

Unfortunately, Oracle 10g is distributed as 64-bit libraries only and the perl version from sunfreeware is a 32-bit executable. If you try to link you get complains about 'wrong ELF class: ELFCLASS64'. The only way to resolve this I found was to recompile perl in 64 bit mode from sources. This is pretty much straight forward, just make sure you include full 64-bit support.

If file \$(which perl) gives you:

```
/usr/bin/perl: ELF 64-bit MSB executable SPARCV9 Version 1 [...
blablabla ...]
```

perl.dbi.users: Compiling and running DBD for Oracle 10g without installing Oracle

you're set. If however it returns:

```
/usr/bin/perl: ELF 32-bit MSB executable SPARC Version 1 [... blablabla ...]
```

it's compilation time.

With a 64-bit version of perl, compiling DBD-Oracle poses no problems, continue at step two.

As usual on Solaris, don't forget to include in you path /usr/ccs/bin so that you can access make and the like.

Step three: Installing Oracle 10g instant Client

See the instructions by Oracle for this step. The instant client packages contain only a few libraries. It doesn't set up your oracle environment, so you have to create your \$ORACLE_HOME directory on your own.

If you'd like to use a tnsnames.ora file for name resolution, you should either create them in \$ORACLE_HOME/network/admin and set \$ORACLE_HOME accordingly or set \$TNS_ADMIN properly.

You also should make sure, that your programs find the oracle libraries by setting \$LD_LIBRARY_PATH properly.

If you installed sqlplus, you now should be able to access the databases of your choice.

Step four: Running DBI programs

If DBD has been properly compiled and the instant client is properly installed, you now can access Oracle databases with DBI from you perl programs.

I hope, these hints help some of you out there.