

RE: [dbi] DBD::ODBC, unixODBC, FreeTDS, MS-SQL, lazy DL, dbd_db_login/SQLSetConnectOption err=-2

Source: <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2005-01/0007.html>

From: Martin J. Evans (martin.evans_at_easysoft.com)

Date: 01/04/05

Date: Tue, 04 Jan 2005 11:58:50 -0000 (GMT)

To: dbi-users@perl.org

Error -2 is `SQL_INVALID_HANDLE` and it invariably means (in the case you specify) that a handle generated by one component has been passed to another e.g. `SQLAllocHandle` for the connection handle was returned by the driver manager but when it was passed to another ODBC API it got through to the driver bypassing the driver manager (or vice versa).

If you imagine the process from calling `DBI->connect`:

1. DBI loads `DBD::ODBC` which is dependent on `libodbc.so`
2. call to `SQLAllocHandle` for the environment. `libodbc.so` is loaded so resolved in `unixODBC` driver manager and handle returned.
3. call to `SQLAllocHandle` for connection. As 2.
4. call to `SQLDriverConnect`. `unixODBC` driver manager loads the ODBC driver which also has some/many/all of the ODBC API symbols. The driver manager uses `dlsym` (in this case) to find the addresses of all the ODBC APIs in the driver.

`RTLD_GROUP`

affects how the symbols are resolved (resolve downwards from reference or from the top).

5. app calls `SQLSetConnectAttr`. This is often the first call after a connect.

If `SQLSetConnectAttr` is resolved in the driver manager and the driver manager has the driver addresses all is well as the connection handle was provided by `unixODBC` and it has the driver's connection handle to pass on.

If `SQLSetConnectAttr` is now resolved in the driver `DBD::ODBC` is passing a connection handle provided by `unixODBC` to the driver which will not recognise it.

The problems seem to occur when:

perl.dbi.users: RE: [dbi] DBD::ODBC, unixODBC, FreeTDS, MS-SQL, lazy DL, dbd_db_login/SQLSetConnectOption err=-2

- a) RTLD_GROUP is not used or RTLD_GLOBAL is used
- b) Perl using lazy linking for loading ODBC.so (dependent on libodbc.so)

You can confirm this with unixODBC tracing where I'd expect you to not see the call to SQLSetConnectAttr because it is going to the driver. You can also turn FreeTDS tracing on.

Some earlier versions of unixODBC set the default to not use RTLD_GROUP and some even earlier do not have --enable-rtldgroup. I'd get unixODBC 2.2.10, configure and build it yourself with --enable-rtldgroup=yes, check dlopen is being called correctly and double check you are using the unixODBC you have just built.

Martin

--

Martin J. Evans
Easysoft Ltd, UK
Development

On 04-Jan-2005 Honza Pazdziora wrote:

```
>
> Hello,
>
> I have succesfully installed
>
>      freetds-0.62.3-1.1.fc3.rf      from Dag Wieers's rpm repository
>      unixODBC-2.2.9-42
>      unixODBC-devel-2.2.9-42      from Fedora Core 3's rpm repository
>      DBD::ODBC 1.13                from sources from CPAN
>
> on top of
>
>      perl, v5.8.3 built for i386-linux-thread-multi
>                                     from FC3's perl-5.8.3-5 rpm
>
> The make test did not go especially well but that was caused by me not
> having enough priviledges on the remote MS-SQL. Accessing the remote
> MS-SQL from Linux box is my goal.
>
> My scripts only work if I add PERL_DL_NONLAZY=1 environment setting
> before starting my perl scripts that use DBD::ODBC -> unixODBC ->
> freetds -> MS-SQL. Without the PERL_DL_NONLAZY=1, the connect to the
> remote server fails with
>
>      DBD: dbd_db_login/SQLSetConnectOption err=-2
>
> Searching the web and archives, the problem is documented in
> Easysoft.com's
>
> http://www.easysoft.com/products/9999/faq\_answer.phtml?ID=664&product=2002
>      Why do I get "SQLSetConnectOption err=-2" errors in my Perl
>      scripts
>
> However, adding --enable-rtldgroup=yes to unixODBC's configure nor
> fiddling with flags passed to dlopen in sys_dl_open function did not
> yield change in behaviour -- PERL_DL_NONLAZY=1 makes the connect (and
> everything after that, including character sets, etc.) work, without
> PERL_DL_NONLAZY=1 I get the SQLSetConnectOption err=-2 error.
```

RE: [dbi] DBD::ODBC, unixODBC, FreeTDS, MS-SQL, lazy DL, dbd_db_login/SQLSetConnectOption err=-2

perl.dbi.users: RE: [dbi] DBD::ODBC, unixODBC, FreeTDS, MS-SQL, lazy DL, dbd_db_login/SQLSetConnectOption error

```
>
> I am not sure yet whether the problem really is in Perl, unixODBC or
> DBD::ODBC. However, as it first appeared after I used DBD::ODBC,
> I thought I ask in dbi-users -- does anyone experience the similar
> problem? Or, does anyone use the combination of software mentioned
> above (or similar) to work successfully with MS-SQL from Linux? Is
> there any way to force the PERL_DL_NONLAZY=1 from inside Perl script,
> so that I would put in into ODBC.pm and save myself the trouble of
> wrapping all scripts with the environment setting?
>
> I can of course send straces / other logs I someone things they should
> be usefull.
>
> Thank you for any hint,
>
> --
> -----
> Honza Pazdziora | adelton@fi.muni.cz | http://www.fi.muni.cz/~adelton/
> .project: Perl, mod_perl, DBI, Oracle, large Web systems, XML/XSL, ...
> Only self-confident people can be simple.
```

RE: [dbi] DBD::ODBC, unixODBC, FreeTDS, MS-SQL, lazy DL, dbd_db_login/SQLSetConnectOption error