

## RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

---

*Source:* <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2005-07/msg00209.html>

---

- *From:* [bbecker@xxxxxxxx](mailto:bbecker@xxxxxxxx) (Brian Becker)
  - *Date:* Tue, 26 Jul 2005 08:01:17 -0400
- 

Was this supposed to happen? A large # of tests now fail...

t/07bind.....ok 1/11Can't rebind placeholder 3 at t/ODBCTEST.pm line 228.

# Looks like you planned 11 tests but only ran 2.

# Looks like your test died just after 2.

t/07bind.....dubious

Test returned status 255 (wstat 65280, 0xff00)

DIED. FAILED tests 3-11

Failed 9/11 tests, 18.18% okay

t/08bind2.....ok 1/5# These are tests which rely upon the driver to tell what the parameter type is for the column. This means you need to ensure you tell your driver the type of the column in bind\_col().

t/08bind2.....NOK 3

# Failed test (t/08bind2.t at line 73)

t/08bind2.....NOK 4

# Failed test (t/08bind2.t at line 77)

t/08bind2.....ok 5/5# Looks like you failed 2 tests of 5.

t/08bind2.....dubious

Test returned status 2 (wstat 512, 0x200)

DIED. FAILED tests 3-4

Failed 2/5 tests, 60.00% okay

t/09multi.....ok

t/20SqlServer....ok 1/37Can't rebind placeholder 2 at t/20SqlServer.t line 108.

# Looks like you planned 37 tests but only ran 2.

# Looks like your test died just after 2.

t/20SqlServer....dubious

Test returned status 255 (wstat 65280, 0xff00)

DIED. FAILED tests 3-37

Failed 35/37 tests, 5.41% okay

-----Original Message-----

From: Martin J. Evans [<mailto:martin.evans@xxxxxxxxxxxxxx>]

Sent: Tuesday, July 26, 2005 6:50 AM

To: Brian Becker

Cc: dbi-users@xxxxxxxx

Subject: RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

Brian,

OK, I've found this now. It is rather more complicated than it looks.

I did this in odbctest (directly to MS SQL Server driver):

```
create table mjenull(a varchar(20))
insert into table values ('AAA')
select * from table where a = isnull(NULL, 'AAA')
```

correctly returns 1 row.

now:

```
SQLPrepare(select * from table where a = isnull(?, 'AAA')
SQLBindParameter(stmt, 1, SQ_C_CHAR, SQL_VARCHAR, 1, 0, NULL, 0, ptr)
NOTE the 1 for ColumnSize
*ptr = SQL_NULL_DATA
SQLExecute
```

no rows returned.

repeat with ColumnSize = 20

```
SQLBindParameter(stmt, 1, SQ_C_CHAR, SQL_VARCHAR, 20, 0, NULL, 0, ptr)
*ptr = SQL_NULL_DATA
SQLExecute
```

1 row correctly returned.

now insert:

```
insert into table values ('A')
```

repeat with ColumnSize = 1 (as DBD::ODBC does):

```
SQLPrepare(select * from table where a = isnull(?, 'AAA')
SQLBindParameter(stmt, 1, SQ_C_CHAR, SQL_VARCHAR, 1, 0, NULL, 0, ptr)
NOTE the 1 for ColumnSize
*ptr = SQL_NULL_DATA
SQLExecute
```

1 row returned and it is the row containing 'A'

I'd guess this is because the SQL Server ODBC driver is looking at your SQLBindParameter call to determine the type and size (1) and hence is saying isnull is going to return 1 character -> 'A' in this case, even though we said 'AAA'.

The solution is to make sure the ColumnSize is what SQLDescribeParam returns.

The diff to DBD::ODBC 1.13 below is a total hack to make this work - it is not

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

complete because it doesn't handle drivers that don't have  
SQLDescribeParam  
(and it is a hack). I'm afraid I don't have time right now to do this  
properly.

```
--- dbdimp.h Mon Oct 11 20:20:55 2004
+++ dbdimp.h.mine Tue Jul 26 10:57:05 2005
@@ -139,6 +139,7 @@
SWORD tgt_sql_type; /* the PH SQL type the
stmt
expects */
SDWORD tgt_len; /* size or precision the stmt
expects */
SDWORD cbValue; /* length of returned value OR
SQL_NULL_DATA */
+ UDWORD cbColDef; /* ColumnSize returned from
SQLDescribeParam */
SDWORD *indics; /* ptr to indicator array for
param
arrays */
int is_array; /* TRUE => parameter array */
```

```
--- dbdimp.c Fri Nov 5 03:19:36 2004
+++ dbdimp.c.mine Tue Jul 26 11:25:41 2005
@@ -2532,6 +2532,7 @@
/* fall through */
/* return 0; */
} else {
+ phs->cbColDef = dp_cbColDef;
if (ODBC_TRACE_LEVEL(imp_sth) >=5)
PerlIO_printf(DBIc_LOGPIO(imp_dbh),
" SQLDescribeParam %s: SqlType=%s,
ColDef=%d\n",
@@ -2747,7 +2748,7 @@
}
cbColDef = phs->biggestparam;
}
-
+
if (!SvOK(phs->sv)) {
/* if is_inout, shouldn't we null terminate the buffer and send
* it, instead?? */
@@ -2766,6 +2767,7 @@
} else {
rgbValue = NULL;
phs->cbValue = SQL_NULL_DATA;
+ cbColDef = phs->cbColDef;
}
}
else {
```

Martin

--

Martin J. Evans  
Easysoft Ltd, UK  
Development

On 26-Jul-2005 Martin J. Evans wrote:

```
> Brian,
>
> Looks like the NULL is getting to the SQL Server
> ODBC driver to me - see below.
>
> Brian Becker wrote:
>> Yes I am referring to the ODBC-ODBC Bridge.
>>
>> Here is the esoobclient.log
>>
> <snipped>
>
>> SQLBindParameter(2bfd88,1,1(Input::),1,12,1,0,0,0,2c17e8)
>> ^oobc_expand_desc_recs(2bfe98,1)
>> Expanding 0, n_descrec=0, expanding to 2
>> ^-oobc_expand_desc_recs()=2c18a4
>> ^oobc_expand_desc_recs(2bff8,1)
>> Expanding 0, n_descrec=0, expanding to 2
>> ^-oobc_expand_desc_recs()=2c19e4
>> Passthru: 1
>> -SQLBindParameter(...)=0
> parameter bound as a SQL_C_CHAR
>
>> SQLExecute(2bfd88)
>> ^put_bound_parameters(2bfd88,27a4f0,1)
>> Driver supports SQLNumParams : 1
>> SQLNumParams returns 1 parameters
>> Processing parameter 1 of 1(1)
>> Initial DataPtr=0, OctetLengthPtr=2c17e8(-1),
>> IndicatorPtr=2c17e8(-1)
>> No DataPtr - looking at indicator
>> Indicator = -1
>> -^put_bound_parameters()=0
> and the indicator says it is a NULL.
>
> <snipped>
>
>> SQLFetch(2bfd88)
>> ^fetch_bound_columns(2bfd88,27a4f0)
>> ^fetch_column_descriptions(27a4f0,2bfd88,ffbf086)
>> -^fetch_column_descriptions()=SQL_SUCCESS (highest result set
```

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

```
>> column = 14)
> <snipped>
>
> and SQLFetch returns rows.
>
>
>> SQLPrepare(2c6558,2c9210,69)
>> SQLPrepare:
>>
>> select * from ORders where CustomerID = isnull(?,'VINET')
>>
>>
>
>> SQLBindParameter(2c6558,1,1(Input::),1,12,1,0,0,0,2c9290)
>> ^oobc_expand_desc_recs(2c6668,1)
>> Expanding 0, n_descrec=0, expanding to 2
>> ^-oobc_expand_desc_recs()=2c934c
>> ^oobc_expand_desc_recs(2c7ad0,1)
>> Expanding 0, n_descrec=0, expanding to 2
>> ^-oobc_expand_desc_recs()=2c948c
>> Passthru: 1
>> -SQLBindParameter(...)=0
> as before
>
>> SQLExecute(2c6558)
>> ^put_bound_parameters(2c6558,27a4f0,1)
>> SQLNumParams returns 1 parameters
>> Processing parameter 1 of 1(1)
>> Initial DataPtr=0, OctetLengthPtr=2c9290(-1),
>> IndicatorPtr=2c9290(-1)
>> No DataPtr - looking at indicator
>> Indicator = -1
>> -^put_bound_parameters()=0
>> -SQLExecute(...)=0
>
> parameter is a NULL.
>
> <snipped>
>
>
>> SQLFetch(2c6558)
>> -SQLFetch(...)=100
> no rows returned.
>
> In both cases I'd say the NULL is getting to SQL Server.
> If you get hold of odbctest which used to be in the ODBC SDK
> you should be able to run this directly to the MS SQL Server
> driver on Windows. If I can squeeze some time today I'll
> try it here.
>
> Martin
```

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

>  
>>  
>> -----Original Message-----  
>> From: Martin J. Evans [<mailto:martin.evans@xxxxxxxxxxxxx>]  
>> Sent: Monday, July 25, 2005 12:59 PM  
>> To: dbi-users@xxxxxxxxx  
>> Subject: RE: MsSQL DBD::ODBC IsNull and undef  
>>  
>> Brian,  
>>  
>> I presume by "Easysoft" you mean Easysoft ODBC-ODBC Bridge?  
>>  
>> If so, can you run your test script again but creating an OOB log  
file  
>> which  
>> you can do by:  
>>  
>> 1. cd /tmp  
>> 2. create odbc.ini file containing only:  
>>  
>> {Settings}  
>> Logging = 0xfffff  
>>  
>> NOTE, they are curly braces.  
>>  
>> 3. WITHOUT changing directory, run you Perl script  
>>  
>> perl /path\_to\_my\_script/xxx.pl  
>>  
>> 4. send me the resulting esoobclient.log\_<PID> which will appear in  
/tmp  
>> (where  
>> <PID> is the process ID).  
>>  
>> Martin  
>> --  
>> Martin J. Evans  
>> Easysoft Ltd, UK  
>> Development  
>>  
>>  
>> On 25-Jul-2005 Brian Becker wrote:  
>>  
>>> I am trying to bind an undef into an IsNull on a varchar field. When  
>>  
>>> I  
>>  
>>> write the query binding directly into the IsNull I get no records,  
yet  
>>> if I create a SQL variable, bind the undef to that, and then use the  
>>

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

```
>> SQL
>>
>>>variable – I get results. I have created a test script that
>>
>> duplicates
>>
>>>the problem (sorry the test is not the best but it does illustrate
the
>>>problem). The test script is done off of the Northwind database for
>>
>> SQL
>>
>>>Server 2000.
>>>
>>>Software:
>>>Easysoft
>>>unixODBC–2.2.11
>>>perl, v5.8.6 built for sun4–solaris
>>>DBD–ODBC–1.13
>>>DBI–1.48
>>>SQL Server 2000
>>>
>>>Test Script:
>>>
>>>#!perl -w -I./t
>>>
>>>use Test::More;
>>>$| = 1;
>>>
>>>use_ok('DBI', qw(:sql_types));
>>>use_ok('ODBCTEST');
>>>
>>># to help ActiveState's build process along by behaving (somewhat) if
a
>>>dsn is not provided
>>>BEGIN {
>>> if (!defined $ENV{DBI_DSN}) {
>>> plan skip_all => "DBI_DSN is undefined";
>>> } else {
>>> plan tests =>4;
>>> }
>>>}
>>>
>>>
>>>#DBI->trace(2);
>>>my $dbh = DBI->connect();
>>>unless($dbh) {
>>># BAILOUT("Unable to connect to the database ($DBI::errstr)\nTests
>>>skipped.\n");
>>> exit 0;
>>>}
```

```
>>>
>>>$dbh->{AutoCommit} = 1;
>>>
>>>#### testing a simple select
>>>
>>>my $rc = 0;
>>>my $CustomerID;
>>>my $sql="
>>> Declare \@CustomerID varchar(20)
>>> SET \@CustomerID=?
>>> select * from ORders where CustomerID =
>>>isnull(\@CustomerID,'VINET')
>>> ";
>>>my $sth = $dbh->prepare(qq{$sql});
>>>$sth->bind_param(1,$CustomerID);
>>>$sth->execute();
>>>my $RowCount=0;
>>>while(my $row = $sth->fetchrow_hashref())
>>>{
>>> $RowCount++;
>>>}
>>>is($RowCount,5, "This works");
>>>
>>>my $sql2="
>>> select * from ORders where CustomerID = isnull(?,'VINET')
>>> ";
>>>my $sth2 = $dbh->prepare($sql2);
>>>$sth2->bind_param(1,$CustomerID);
>>>$sth2->execute();
>>>$RowCount=0;
>>>while(my $row = $sth2->fetchrow_hashref())
>>>{
>>> $RowCount++;
>>>}
>>>is($RowCount,5, "This Doesn't");
>>># Test(1);
>>># clean up
>>>$sth->finish
>>>exit(0);
>>>
>>>__END__
>>>
>>>
>>>COMMAND OUTPUT:
>>>
>>>#perl Makefile.PL
>>>Useless use of private variable in void context at Makefile.PL line
>>
>> 431.
>>
>>>Configuring DBD::ODBC ...
```

```
>>>
>>>
>>>>> Remember to actually *READ* the README file!
>>>
>>> And re-read it if you have any problems.
>>>
>>>Using DBI 1.48 (for perl 5.008006 on sun4-solaris) installed in
>>>/usr/perl5.8.6/lib/site_perl/5.8.6/sun4-solaris/auto/DBI/
>>>Using ODBC in /usr/local/easysoft/unixODBC
>>>
>>>Umm, this looks like a unixodbc type of driver manager.
>>>We expect to find the sql.h, sqlext.h and (which were
>>>supplied with unixODBC) in $ODBCHOME/include directory alongside
>>>the /usr/local/easysoft/unixODBC/lib/libodbc.so library. in
>>>$ODBCHOME/lib
>>>
>>>
>>>Checking if your kit is complete...
>>>Looks good
>>>Injecting selected odbc driver into cc command
>>>Injecting selected odbc driver into cc command
>>>Using DBI 1.48 (for perl 5.008006 on sun4-solaris) installed in
>>>/usr/perl5.8.6/lib/site_perl/5.8.6/sun4-solaris/auto/DBI/
>>>Writing Makefile for DBD::ODBC
>>>
>>>The DBD::ODBC tests will use these values for the database
connection:
>>> DBI_DSN=dbi:ODBC:dsn_Northwind e.g. dbi:ODBC:demo
>>> DBI_USER=webuser
>>> DBI_PASS=test
>>>
>>>#make
>>>cp Changes blib/lib/DBD/ODBC/Changes.pm
>>>cp ODBC.pm blib/lib/DBD/ODBC.pm
>>>/usr/bin/perl -p -e "s/~DRIVER~/ODBC/g"
>>>/usr/perl5.8.6/lib/site_perl/5.8.6/sun4-solaris/auto/DBI/Driver.xst
>
>>>ODBC.xsi
>>>/usr/bin/perl /usr/perl5.8.6/lib/5.8.6/ExtUtils/xsubpp -typemap
>>>/usr/perl5.8.6/lib/5.8.6/ExtUtils/typemap ODBC.xs > ODBC.xsc && mv
>>>ODBC.xsc ODBC.c
>>>Warning: duplicate function definition 'data_sources' detected in
>>>ODBC.xs, line 202
>>>gcc -c -I/usr/local/easysoft/unixODBC/include -I.
>>>-I/usr/perl5.8.6/lib/site_perl/5.8.6/sun4-solaris/auto/DBI/
>>>-fno-strict-aliasing -pipe -I/usr/local/include -D_LARGEFILE_SOURCE
>>>-D_FILE_OFFSET_BITS=64 -O -DVERSION="1.13\"
-DXS_VERSION="1.13\"
>>>-fPIC "-I/usr/perl5.8.6/lib/5.8.6/sun4-solaris/CORE"
>>>-I/usr/local/easysoft/unixODBC/include ODBC.c
>>>gcc -c -I/usr/local/easysoft/unixODBC/include -I.
```

RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef

```
>>>-I/usr/perl5.8.6/lib/site_perl/5.8.6/sun4-solaris/auto/DBI/
>>>-fno-strict-aliasing -pipe -I/usr/local/include -D_LARGEFILE_SOURCE
>>>-D_FILE_OFFSET_BITS=64 -O -DVERSION=\"1.13\"
-DXS_VERSION=\"1.13\"
>>>-fPIC "-I/usr/perl5.8.6/lib/5.8.6/sun4-solaris/CORE"
>>>-I/usr/local/easysoft/unixODBC/include dbdimp.c
>>>Running Mkbootstrap for DBD::ODBC ()
>>>chmod 644 ODBC.bs
>>>rm -f blib/arch/auto/DBD/ODBC/ODBC.so
>>>LD_RUN_PATH="/usr/local/easysoft/unixODBC/lib" gcc -G
>>
>> -L/usr/local/lib
>>
>>>ODBC.o dbdimp.o -o blib/arch/auto/DBD/ODBC/ODBC.so
>>>-L/usr/local/easysoft/unixODBC/lib -lodbc
>>>chmod 755 blib/arch/auto/DBD/ODBC/ODBC.so
>>>cp ODBC.bs blib/arch/auto/DBD/ODBC/ODBC.bs
>>>chmod 644 blib/arch/auto/DBD/ODBC/ODBC.bs
>>>Manifying blib/man3/DBD::ODBC.3
>>>
>>>Then of course make test fails on my test script above.
>>>
>>>Brian Becker
>>>Jaeb Center for Health Research
>>>15310 Amberly Drive
>>>Suite 350
>>>Tampa, FL 33647
>>>tel: 813-975-8690
>>>fax: 813-975-8761
>>>email: bbecker@xxxxxxxxx
>>>http://www.jaeb.org
>>>
>>>
.
```

---

• **Follow-Ups:**

◆ **Re: MsSQL DBD::ODBC IsNull and undef**

◇ From: Martin J. Evans

• Prev by Date: **RE: [dbi] Re: MsSQL DBD::ODBC IsNull and undef**

• Next by Date: **Re: MsSQL DBD::ODBC IsNull and undef**

• Previous by thread: **MsSQL DBD::ODBC IsNull and undef**

• Next by thread: **Re: MsSQL DBD::ODBC IsNull and undef**

• Index(es):

◆ **Date**

◆ **Thread**