

Sybase module 0.37pre2 released

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-03/3238.html>

From: Dave Cole (djc_at_object-craft.com.au)

Date: 03/21/05

Date: Mon, 21 Mar 2005 10:15:36 +1100

To: python-sybase@object-craft.com.au

WHAT IS IT:

The Sybase module provides a Python interface to the Sybase relational database system. It supports all of the Python Database API, version 2.0 with extensions.

NOTES:

This release contains a number of small bugfixes and patches received from users.

I have been unable to find the source of the memory leak reported here:

<http://www.object-craft.com.au/pipermail/python-sybase/2004-December/000346.html>

The test program I wrote follows:

```
-----
import sys
import Sybase

db = Sybase.connect(..., auto_commit=True)
db.execute("""
if exists (select name from sysobjects where name = "sp_test_leak")
begin
    drop procedure sp_test_leak
end
""")
db.execute("""
create procedure sp_test_leak
    @arg int
as
    select @arg
""")
for i in range(200):
    for j in range(1000):
        c = db.cursor()
        c.callproc('sp_test_leak', {'@arg': 12345 })
```

comp.lang.python: Sybase module 0.37pre2 released

```
sys.stdout.write('%3d\r' % i)
sys.stdout.flush()
sys.stdout.write('\n')
```

If someone is able to modify this and come up with a leaking result I am interested in working on the fix.

You can build for FreeTDS like this:

```
python setup.py build_ext -D HAVE_FREETDS -U WANT_BULKCOPY
python setup.py install
```

The module is available here:

<http://www.object-craft.com.au/projects/sybase/download/sybase-0.37pre1.tar.gz>

The module home page is here:

<http://www.object-craft.com.au/projects/sybase/>

CHANGES SINCE 0.36:

- * Cursor state initialisation fix from Skip Montanaro.
- * Callback declaration fix on Windows from Vadim Beloborodov.
- * Cursor output parameters now work when parameters are passed as a sequence.
- * Output parameters now work for FreeTDS 0.62.4.

```
1> create procedure sp_test_output
2> @num int, @result int output
3> as
4> select @result = @num
5> go
```

```
params = c.callproc('sp_test_output', {'@num': 12345,
                                       '@result': Sybase.OUTPUT(1)})
print params['@result']
```

- * The CS_STATUS_RESULT result set is now consumed internally in the Cursor and does not appear in the result sets consumed by the fetch and nextset methods.

The return value from the stored procedure is available in the .return_status member of the Cursor. It will only contain a meaningful value once all of the row result sets have been consumed.

comp.lang.python: Sybase module 0.37pre2 released

Note that this does not work with FreeTDS 0.62.4. The return_status seems to always be 0. Research shows that the problem is probably in the CT emulation layer as tsql displays the correct value, but sqsh displays 0.

- * Output hook patch from Ty Sarna has been applied.
- * Applied patch from Andre Sedinin to improve error handling.
- * Improved detection of SYBASE_OCS.

– Dave

--

<http://www.object-craft.com.au>