

PLS-00306 when calling an Oracle package procedure through a private synonym that returns a cursor.

Source: <http://www.tech-archive.net/Archive/Data/microsoft.public.data.ado/2004-12/0203.html>

From: Roel Schreurs (schreurs_roel_at_hotmail.com)

Date: 12/17/04

Date: 17 Dec 2004 05:10:11 -0800

Hi All,

I am using ADODB to retrieve data from an Oracle 8.1.7. server. I have a package that contains a procedure that returns a REF CURSOR. In my client code I use a ADODB.Command object to execute the procedure and a ADODB.RecordSet object to access the results.

Now I want to use a private synonym to be able to access the package from another schema. This fails with the following error:
PLS-00306: wrong number or types of arguments in call to '<Procedure>'.

A function in the package can be executed through the synonym without problems.
An SQL script (see 6.) succeeds when executed using a SQL tool (Toad).

Below are the steps to reproduce this behaviour.

I suppose this has something to do with the way the OraOLEDB provider treats the cursor that is returned as an out parameter.

Any help would be appreciated.

Roel Schreurs

1. Package

```
-----  
CREATE OR REPLACE PACKAGE TestPackage  
AS
```

```
    FUNCTION GetValue  
    RETURN VARCHAR2;
```

```
    TYPE ValueRecType IS RECORD
```

PLS-00306 when calling an Oracle package procedure through a private synonym that returns a cursor.

microsoft.public.data.ado: PLS-00306 when calling an Oracle package procedure through a private synonym that returns a

```
( Value VARCHAR2(10));
```

```
TYPE ValueCurType IS REF CURSOR RETURN ValueRecType;
```

```
PROCEDURE GetValueCur  
( cur OUT ValueCurType);
```

```
END TestPackage;
```

2. Package body

```
CREATE OR REPLACE PACKAGE BODY TestPackage  
AS
```

```
FUNCTION GetValue  
RETURN VARCHAR2  
IS  
BEGIN  
    RETURN 'Successful!';  
END;
```

```
PROCEDURE GetValueCur  
( cur OUT ValueCurType)  
IS  
BEGIN  
    OPEN cur FOR  
    SELECT 'Successful!' AS "Value"  
    FROM DUAL;  
END;
```

```
END TestPackage;
```

3. Synonym

```
create synonym TestSynonym for TestPackage
```

4. Demo ASP that fails. Please adjust <data source>, <user> and <password>.
(includes commented-out alternatives that work.)

```
<% Option Explicit  
Response.Expires = 0%>  
<html>
```

```
<head>  
</head>
```

```
<body>
```

```
<center>  
<h1>Synonym tester</h1>
```

PLS-00306 when calling an Oracle package procedure through a private synonym that returns a cursor.

microsoft.public.data.ado: PLS-00306 when calling an Oracle package procedure through a private synonym that returns a

```
<%
  Dim objConnection
  Dim objCommand
  Dim objRecordset
  Dim DBError

  Set objConnection = Server.CreateObject("ADODB.Connection")
  With objConnection
    .ConnectionString = "Provider=OraOLEDB.Oracle;" & _
      "Data Source=" & "<data source>" & ";" & _
      "User ID=" & "<user>" & ";" & _
      "Password=" & "<password>" & ";" & _
      "PLSQLSet=1;enlist=false;"
    .Open
  End With

  Set objCommand = Server.CreateObject("ADODB.Command")
  Set objRecordset = Server.CreateObject("ADODB.RecordSet")

  With objCommand
    Set .ActiveConnection = objConnection
    .CommandType = 1
    'These three will work.
    '.CommandText = "SELECT testpackage.GetValue AS Value from
dual"
    '.CommandText = "{call testpackage.GetValueCur}"
    '.CommandText = "SELECT testsynonym.GetValue AS Value from
dual"
    'This one will fail.
    .CommandText = "{call testsynonym.GetValueCur}"
    Set objRecordset = .Execute()
  End With

  Response.Write "Selected Value: " &
objRecordSet.Fields("Value").Value
%>
</center>
</body>
</html>
```

5. Error message

```
-----
OraOLEDB (0x80040E14)
ORA-06550: line 1, column 7: PLS-00306: wrong number or types of
arguments in call to 'GETVALUECUR' ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
Test.asp, line 42
-----
```

6. SQL Script that works fine.

```
-----
BEGIN
```

PLS-00306 when calling an Oracle package procedure through a private synonym that returns a cursor.

microsoft.public.data.ado: PLS-00306 when calling an Oracle package procedure through a private synonym that returns a

```
DECLARE ValueRec TestSynonym.ValueRecType;
        ValueCur TestSynonym.ValueCurType;
BEGIN
    TestSynonym.GetValueCur(ValueCur);
    FETCH ValueCur INTO ValueRec;
    DBMS_OUTPUT.PUT_LINE('Selected value: ' || ValueRec.Value);
END;
END;
-----
```