

# Re: Passing Back Table Information from Oracle .NET Stored Procedures

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-07/msg03878.html>

---

- *From:* "G.Doten" <[gdoten@xxxxxxxx](mailto:gdoten@xxxxxxxx)>
  - *Date:* Tue, 31 Jul 2007 08:34:53 -0400
- 

Mr. Arnold wrote:

<[jehugaleahsa@xxxxxxxx](mailto:jehugaleahsa@xxxxxxxx)> wrote in message  
[news:1185809768.963317.39600@xx](mailto:news:1185809768.963317.39600@xx)

You are still not understanding my question. I am not trying to call a stored procedure, not in the typical PL/SQL sense. Okay, it is possible to upload a .NET assembly to an Oracle database. In doing so, the database engine can execute the .NET assembly under a context. In a sense, you are executing a .NET assembly just like you would any other PL/SQL stored procedure.

This is not PL/SQL. I am not trying to get a ref cursor. I am trying to get the results of a select command from a ".NET stored procedure". Check out this link if you have no clue what I am talking about:  
[http://www.oracle.com/technology/pub/articles/mastering\\_dotnet\\_oracle/williams\\_sps.html](http://www.oracle.com/technology/pub/articles/mastering_dotnet_oracle/williams_sps.html).

If you take the time to look at the link, you will know exactly what I am talking about. All the example links I have been shown have been for working with ADO .NET or writing PL/SQL. I just want to get a result set from a .NET stored procedure. And, yes, it *\*is\**, really, truly, a .NET stored procedure.

So? Even in the example you are providing, it's returning a result set. I don't care if it's MS SQL in-line code or Oracle in-line code, MS SQL Server, Oracle Proc, or this so called .NET Stored Procedure using a Select statement, they are all returning a RESULT SET whether it be one row or multiple rows.

You put the RDR into a loop until it hits EOF and read the data in the result set, row by row.

CountryName = rdr.GetString(0); // in the example is CountryName field on the row.

Or CountryName = rdr.GetString("CountryName"); // or something like that.

## Re: Passing Back Table Information from Oracle .NET Stored Procedures

In either of above the cases, you must use a `DATAREADER` to read the rows and address the fields on the rows of a returned result set.

I don't think that the sample pointed to above on the Oracle site is returning a result set. It looks to me like the `GetCountryName` stored procedure is returning a one column row called `CountryName`. And it doesn't return it in a conventional SQL result set or whatever. To call the example from a regular C# method (a data layer method, say) it would be something like this:

```
string countryName = StoredProcedures.GetCountryName(countryId);
```

jehugaleahsa, to return a result set from a C# stored procedure like the `GetCountryName` one have you tried the `context` object to return the reader?

```
public class StoredProcedures
public static void ReturnDataReader()
{
using (OracleConnection conn = new OracleConnection(...))
{
conn.Open();
OracleCommand cmd = conn.CreateCommand();
cmd.CommandText = "SELECT Foo FROM Table1";
OracleDataReader reader = new OracleDataReader(CommandBehavior.Close);
SqlContext.Pipe.Send(reader);
}
}
}
```

(The above is typed in from memory.) The key is the `SqlContext` and it's `Pipe` method. While I haven't tried this, I think that it causes the C# stored procedure to return the Oracle result set to the caller, just like when you do the same `SELECT` statement from a PL/SQL stored procedure.

I used `SqlContext` above because I don't want to bother registering at the Oracle site to download their Oracle Database Extensions for .NET in order to get at the presumably extant `OracleContext` class.

This is interesting: I had no idea that Oracle had integrated with the CLR as SQL Server did!

I'd be very interested in knowing if the above is correct or not.

--  
-glenn-

.