Accessing OLEDB Connection Managers in a Script

[Matt Masson - MSFT](http://blogs.msdn.com/54640/ProfileUrlRedirect.ashx)

22 Aug 2008 7:05 PM

Accessing ADO.Net Connection Managers from an SSIS script task / script component is pretty easy – you just need to cast the object returned from AcquireConnection() to the appropriate class (i.e. SqlConnection if you’re using SQL Native Client).

SqlConnection conn = (SqlConnection)Dts.Connections["adonet"].AcquireConnection(null);

If you can’t use ADO.Net for some reason, and are using OLEDB connection managers, it’s a little trickier. Since the AcquireConnection() method of the OLEDB connection manager returns a native COM object, I didn’t think there was a way to make this work, but today someone showed me how to do it!

By casting the Connection Manager’s InnerObject to the IDTSConnectionManagerDatabaseParameters100 interface (IDTSxxx90 in 2005), you can call the GetConnectionForSchema() method to return an OleDbConnection object.

**2008 (C#):**

ConnectionManager cm = Dts.Connections["oledb"];

IDTSConnectionManagerDatabaseParameters100 cmParams = cm.InnerObject as IDTSConnectionManagerDatabaseParameters100;

OleDbConnection conn = cmParams.GetConnectionForSchema() as OleDbConnection;

**2005 (VB)**:

Dim cm As ConnectionManager

Dim cmParam As Wrapper.IDTSConnectionManagerDatabaseParameters90

Dim conn As OleDb.OleDbConnection

cm = Dts.Connections("oledb")

cmParam = CType(cm.InnerObject, Wrapper.IDTSConnectionManagerDatabaseParameters90)

conn = CType(cmParam.GetConnectionForSchema(), OleDb.OleDbConnection)

Note, you’ll need to add a reference to the Microsoft.SqlServer.DTSRuntimeWrap assembly to get the IDTSConnectionManagerDatabaseParameters100 interface. If you’re doing this in a script task, you’ll need to prefix the Microsoft.SqlServer.Dts.Runtime.Wrapper namespace (or use fully qualified names) so that it doesn’t conflict with the namespace for the VSTA proxy classes.

Keep in mind that there are a couple of limitations with this approach:

1. You won’t be able to enlist in the current transaction
2. This connection doesn’t honor the “retain same connection” setting

ADO.Net is still the recommended connection manager type for scripts, but I found this to be a nice work around.

**[Comments](http://blogs.msdn.com/b/mattm/rsscomments.aspx?WeblogPostID=8889077)**

* [**dtisdale**](http://blogs.msdn.com/188440/ProfileUrlRedirect.ashx)

16 Mar 2009 4:16 PM

Could you please explain what you did to access the DTSRuntimeWrap assembly?  I'm trying to programmatically get information about an OLEDB provider that I used to connect to a SAS datasource, so ADO.NET is not an option for me.

* **[http://i1.social.s-msft.com/profile/u/avatar.jpg?displayname=matt+masson+-+msft&size=large](http://blogs.msdn.com/54640/ProfileUrlRedirect.ashx)**[**Matt Masson - MSFT**](http://blogs.msdn.com/54640/ProfileUrlRedirect.ashx)

16 Mar 2009 4:25 PM

In 2008, you should be able to simply add it as an assembly reference for your project (DTSRuntimeWrap can be found in the GAC).

In 2005, you'll need to add the assembly to the .NET framework directory. There is a good writeup on how to do this on sqlservercentral.com.

<http://www.sqlservercentral.com/articles/SQL+Server+2005+-+SSIS/3180/>