

# Re: SQLDMO with VB6

---

*Source:*

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.tools/2007-10/msg00095.html>

---

- *From:* MylesJ <[MylesJ@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:MylesJ@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 19 Oct 2007 14:51:01 -0700
- 

Thanks for the reply, Tibor. Unfortunately, I'd pretty much been through all that. I think I've discovered the missing file: 80\tools\binn\msvcr71.dll. I observed that the MS Backward Compatibility package added this file (among others) to my SQL Server directory. I added it to my project, and now SQLDMO is running fine...at least so far. I'd say MS might consider adding msvcr71.dll to the list in article 326613 on how to distribute SQL-DMO.

"Tibor Karaszi" wrote:

We played with re-distributing DMO, and I think we finally got it working. I wasn't the one doing the work, but I was partly involved. I recall that some KB didn't have the correct information, so we had to tweak this a bit. This was perhaps 6-7 years ago, so I don't recall all details. I looked around in my documents, and I've pasted below the notes from a txt file I found. Not much to go on, I'm afraid, but it might give you some leads...

=====  
8.0, from REDIST.TXT  
=====

sqldmo.dll Distributed Management Objects COM  
sqldmo.rll Distributed Management Objects Resource File  
sqlresld.dll SQL Enterprise Manager Resource DLL Loader  
sqlsvc.dll Database Service Layer  
sqlsvc.rll Database Service Layer Resource DLL  
sqlunirl.dll SQL Server Unicode/ANSI Translation Layer  
w95scm.dll SQL Service Control Manager Abstraction Layer

INSTALLATION NOTES FOR DISTRIBUTED MANAGEMENT OBJECTS (DMO)  
-----

The sqldmo.dll file must be registered using the regsvr32.exe utility.  
Example: regsvr32 80\Tools\Binn\sqldmo.dll

The sqlunirl.dll file should reside in the system folder (i.e., \winnt\system32 or \windows\system).

=====  
7.0, from Q248241  
=====

#### SUMMARY

This article documents the steps necessary to enable SQL Server Distributed Management Objects (SQL-DMO) client side functionality without the need to install the SQL Server Client Utilities.

#### MORE INFORMATION

##### Typical Scenario

You write an application that takes advantage of the SQL-DMO object model in SQL Server. Your application runs fine on the server computer itself, and it runs fine on client computers when the SQL Server Client Utilities are installed. However, by itself, the application does not run. Additionally, you may see several errors related to this as well. The most common errors are:

ActiveX can't create object

Field is not bound correctly

Cannot find entry point

NOTE: Installing Microsoft Data Access Components (MDAC) does not resolve the problem.

MDAC does not install the Sqldmo.dll file or any of the many dependent DLLs associated with the

Sqldmo.dll file. What MDAC does include is many of the related DLLs that a SQL-DMO client connection

uses. Regardless, bundling MDAC with your application installation is not going to be the most

efficient answer either, because you do not need all of the MDAC DLLs to get your DMO connection to

work. Also, MDAC installs many additional features that you will likely never use on your client

computers. The main point here is that you do not need to install all of MDAC to get the functionality you want, just the necessary DLLs. The DLLs are covered later in this article.

Because different types of DLLs and where you need to place them are discussed, let's digress for a

moment and briefly review Win32 and COM based DLLs. For the purposes of this article, all you need

to know is how to tell them apart and then what to do with the DLLs once you identify them.

With

that in mind, here is what you need to know:

You need to register COM based DLLs through the Regsvr32 utility.

You need to place native Win32 DLLs in the application path.

## Re: SQLDMO with VB6

You may also place native Win32 DLLs into the Win32 directory. Sometimes, it may be difficult to make a distinction between COM and non-COM based DLLs; however,

generally a COM based DLL always has the following entry points:

DllGetClassObject

DllRegisterServer

DllUnregisterServer

DllCanUnloadNow

To view the entry points, right-click the DLL, and then click Quick View on the shortcut menu. The

information that appears is noted in the "Export Table" section of the DLL information.

Following is the list of SQL-DMO related DLLs that you need to enable DMO from a client:

Sqldmo.dll

Sqldmo.rll

Sqlsvc.dll

Sqlsvc.rll

Sqlwoa.dll

Sqlresld.dll

Sqlwid.dll

W95scm.dll

From the preceding list of DLLs, the Sqldmo.dll is the only one that you need to register on the

client computer. However, in order to successfully register the Sqldmo.dll file, the Sqldmo.rll file

must be present on the client computer in the following directory:

(server side location): c:\Mssql7\Binn\Resources\1033

(client side location): c:\Winnt\System32\Resources\1033

The Sqldmo.rll file always draws questions. An .rll file is a localized resource file. The resource

directory varies based upon the national language configured on the SQL Server and client install.

In this instance, directory 1033 is a decimal representation of the language identifier 0X0409, indicating U.S. English. After this .rll file is in place, you can register the Sqldmo.dll file by invoking the Regsvr32 utility. The command to register the file is:

```
C:\Regsvr32 Sqldmo.dll
```

Upon successful registration, you should receive a message that states:

```
DLLRegisterServer in SQLDMO.dll succeeded.
```

If you want to automate this registration into your application setup routine, refer to the various

switches associated with the Regsvr32 utility. The command to register the file silently is:

```
c:\regsvr32 /s- sqldmo.dll
```

The directory that contains the Sqldmo.dll file must have a specific structure on the client computer; otherwise you may receive an error message similar to the following:

```
LoadLibrary("C:\Winnt\System32\sqldmo.dll") failed.
```

```
GetLastError returns 0x0000007e
```

Re: SQLDMO with VB6

This error means that error 126 (expressed in decimal), which corresponds to a "specified module not found" error, occurred. In this instance, this relates to the Ssqlmo.rll file. That is, the Regsvr32 utility finds the Ssqlmo.dll file, but cannot complete its registration unless the Ssqlmo.rll file is also available in the expected 1033 subdirectory.

To avoid this, do the following:

Add a directory called "Resources" to the location where the Ssqlmo.dll resides.

Add a subdirectory under Resources called "1033".

In this directory, copy the Ssqlmo.rll file.

When complete, the directory structure should look like:

Directory that contains Ssqlmo.dll\Resources\1033

And that 1033 directory should contain the Ssqlmo.rll file.

In addition to the DLLs mentioned previously, you also want to make sure that you have the proper

netlibrary DLLs installed. The main netlibrary DLLs you need here are:

Named Pipes: Dbnmpntw.dll

Sockets: Dbmssocn.dll

Multi-Protocol: Dbmsrpcn.dll

These are Win32 based DLLs, so you do not need to register these DLLs. Just place the DLLs into the

System32 directory for Microsoft Windows NT or into the System directory for Microsoft Windows 95 or

Windows 98.

After you complete the preceding steps, the client application should start without any of the initial problems caused by the absence of the necessary files.

--

Tibor Karaszi, SQL Server MVP

<http://www.karaszi.com/sqlserver/default.asp>

[http://sqlblog.com/blogs/tibor\\_karaszi](http://sqlblog.com/blogs/tibor_karaszi)

"MylesJ" <MylesJ@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message  
<news:6918CBCC-8136-464F-A8E2-9F3B6F9428A1@xxxxxxxxxxxxxxxxxxxx>

I'm trying to distribute SQLDMO with my VB6 app, which connects to SQL Server

2005 Express. I'm using InstallShield to make my installer. I used the MS article 326613 to get a list of the files I need to distribute. I added those files to my install, and designated SQLDMO.DLL for COM registration.

After

installing, I get an automation error when I try to make a SQLDMO object. If I run the MS SQL Server Backward Compatibility package on the computer, SQLDMO runs fine after that. It looks like the Backward Compatibility package

Re: SQLDMO with VB6

installed some extra files that aren't listed in the KB article cited above. The problem is, I can't expect my end-users to install the Backward Compatibility package, and I can't invoke that msi from inside my installer. Does anybody have a list of what files are installed by the Backward Compatibility package when one chooses only the SQLDMO option? I think the list in the KB article is incomplete. Thanks in advance for any ideas.