

ADO.NET Upgrade FAQ

01/10/2024 1:41 pm EST

Why did we perform this change?

For two reasons. The first reason was that we've been running on extended life support using DBNETLIB.DLL and FreeTDS. We've recently made the change to make SqlBold.dll compatible with newer versions of DBNETLIB.DLL, however support for this library has been deprecated by Microsoft for years, and any future update could remove compatibility. Second, and more importantly, is that this connection method is not TLS 1.2 compliant. A new technology to talk to the database was needed.

Why was ADO.NET chosen?

Of the technologies that Microsoft is currently supporting for database access, only two have TLS 1.2 support: ADO.NET and OLE DB. OLE DB had unreasonable minimum requirements for SQL Server and Windows, plus it is a technology built on the decades-old COM technology, which we didn't want to introduce more of into our technology stack. In addition, building it in ADO.NET means that we can reuse this library as we migrate mode code to .NET.

What does the new technology look like?

The new driver will remove the need for SqlBold.dll and DBNETLIB.DLL completely. There are two new driver files that will need to exist in the Manitou folder.

- Drivers.SqlServer.dll
 - This is the core .NET code that will interface with the database. It is a C# assembly that can be used by .NET applications directly.
- Drivers.SqlWrapper.dll
 - This is an adapter to expose the functionality of BoldTechnologies.Drivers.SqlServer.dll to C++ applications. This library is written in C++/CLI and performs all of the managed to unmanaged transitions as well as data marshaling between the managed and unmanaged worlds. It exposes a basic C interface for legacy applications.
- (Changes in database.lib)
 - A new implementation of clsVendorSpecificDb was added (DB_VENDOR_SQLDOTNET) that talks to the wrapper DLL. This adapts the new driver to work with all the DatabaseAccess, RecordSets, and clsDatabase code.

Should I stop distributing SqlBold.dll?

No. Currently, we're supporting the rollback to SqlBold.dll until we gain more confidence with the new driver. We should be distributing SqlBold.dll for the foreseeable future.

How do I migrate to the new driver?

You'll need to have a new build of all of the C++ components of the system. This includes MMonitor, MSM, Sentry, Broker, AppServer, SignalHandler, FEP, etc. In addition, you'll need `BoldTechnologies.Drivers.SqlServer.dll` and `BoldTechnologies.Drivers.SqlWrapper.dll`.

Once a patch is built, you're free to use the distributor method to upgrade. If you're a developer or want to do it manually follow these steps:

- Stop the entire system
- Close the MSM (leaving this running will cause the monitor to auto-restart before you can upgrade it)
- Stop the monitor in the services control panel
- Upgrade the system with the new code
 - Ensure you've got the two new driver files in your Manitou folder too
- Run up the new MSM
 - The new MSM will automatically migrate the settings in the registry to use the new driver

How can I verify that I'm using the new driver?

If any component of the system is using `SqlBold`, it will log the following in the logger for every database connection it creates (generally on app startup):

WARNING: Using deprecated `SqlBold` database driver!

In addition, you can verify the system has changed to the new setting by checking the registry at:

HKEY_CURRENT_USER\Software\Bold Technologies\Manitou\Manitou System Manager\MSM\Initialization

Each config has a separate set of keys, but the default config 0 should be changed at a minimum. This is the key you're looking for:

Config 0 Main Database Type

If this has a value of 2, you're using `SqlBold`. If it has a value of 6, you're using the new ADO.NET driver.

What if the new driver is so broken I need to roll back to using `SqlBold`?

This option was intended as a last-resort option in production. Please consider working through the problem before rolling back. This driver needs to become as stable and mature as `SqlBold`. Not using it isn't doing anyone any favors.

The system will ensure that your database type is always at 6. This ensures that all sites get migrated to the new database driver. Just changing the type to 2 and restarting the system will just cause it to automatically revert back to 6.

If you need to roll back, you'll need to change one additional new registry setting to confirm you legitimately want to run on database type 2. Here are the steps:

- Stop the entire system
- Close the MSM
- Stop the monitor service in the services control panel
- Open regedit and navigate to HKEY_CURRENT_USER\Software\Bold Technologies\Manitou\Manitou System Manager\MSM\Initialization
- Edit 'Config X Allow Legacy Drivers' to be 1
- Edit Config X Main Database Type' to be 2
- Start the MSM and restart the system

Okay, I've rolled back to SqlBold, but want to now start using the ADO.NET driver again. What do I do?

Basically, the inverse of the rollback:

- Stop the entire system
- Close the MSM
- Stop the monitor service in the services control panel
- Open regedit and navigate to HKEY_CURRENT_USER\Software\Bold Technologies\Manitou\Manitou System Manager\MSM\Initialization
- Edit 'Config X Allow Legacy Drivers' to be 0
- Edit Config X Main Database Type' to be 6
- Start the MSM and restart the system