Paper 1215-2017

Best Practices in Connecting External Databases to SAS®

Mathieu Gaouette, Videotron S.E.N.C.

ABSTRACT

Connecting database schemas to libraries in the SAS® metadata is a very important part of setting up a functional and useful environment for business users. This task can be quite difficult for the untrained administrator. This paper addresses the key configuration items that often go unnoticed but can make a big difference. Using the wrong options can lead to poor database performance or even to a total lockdown, depending on the number of connections to the database.

INTRODUCTION

When I started administrating my first SAS® environment via the SAS management console®, I quickly realized that connecting external databases could be tricky. Although setting up the server definition is fairly easy, I have found that setting up libraries was a whole different ball game. The library creation wizard is fairly simple but the advance panel tab shows multiple options that vary from one DBMS to another. In a lot of cases, ignoring these options and using the default parameters will lead to performance issues.

For years now I have been on a quest to find the best practices when it comes to connecting external databases especially for the library definition portion. After a while I realized that I was better off establishing a set of best practices and then work with other experts to refine it.

What you will find in these pages is the result of years of experience dealing with external database connections. It is my humble contribution to a foundation of best practices for DBMS connections within the SAS management console®. Because of the simple nature of configuring the server portion in the SAS management console®, it will not be covered here. Let's start by stating that whenever possible, native connectors should be used as opposed to ODBC connectors. Using native connectors generally allows SAS® to do more with your DBMS.

This paper will help the administrator identify possible performance problems in establishing connections to external databases and avoid them through optimal configuration. It can also help the end user understand how SAS® handles queries to external databases and help them optimize their code.

SETTING UP THE BASE

When a server connection has been defined in the server manager section of the SAS Management Console®, the base library definition can be laid down. Launching the library definition wizard can be done with a right click on "Libraries" under the "Data Library Manager" group.

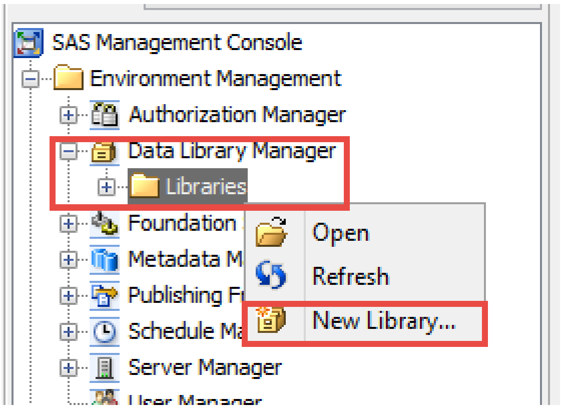


Figure 1. Launching the New Library wizard

Setting up that first library is straight forward with one exception, the server that can access your library. This question comes in right after you provided the library type (Base, Oracle …). With a standard or default install, there are two servers to pick from:

- SASMeta
- SASApp

If you are new to the SAS® administration and the whole SAS® Metadata world, chances are that you don't really know what the difference is between these two. If so, then you should go with SASApp. This is the one that most users will go through with their regular usage (SAS/Enterprise Guide®, SAS/Enterprise Miner® …).

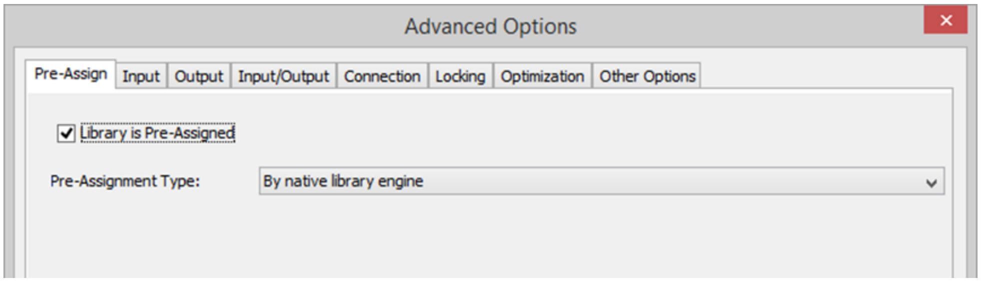
OPENING PANDORA'S BOX

With a new library setup using the basic options provided with the wizard, the foundation for our working library has been established. In most cases, this library will be working for users right away. However, setting up the advanced options is crucial to ensure proper performance and functionality.

Accessing the advanced options is done by editing the new library's properties. This section can also be accessed during the initial configuration of the library. It is recommended to proceed in two steps and follow an iterative approach. It is also good to note that tweaking advanced options during the initial setup have produced some errors with older version of the SAS Management Console®.

PRE-ASSIGNED LIBRARIES

The first choice we are presented with in the advanced options panel is whether or not we want our library to be pre-assigned. A library that is not pre-assigned forces the users to establish the connection before using it. For this paper, we will focus solely on pre-assigned libraries.



☑ Library is Pre-Assigned

Pre-Assignment Type: By native library engine

Figure 2. Advanced options: Pre-Assign tab

The second option to tweak is the Pre-Assignment type. It is really important to know the difference between the possible values and understand SAS®' behavior in each case.

- By native library engine

  This is the default choice. For DBMS libraries, it lets the database handle the security. For connections to external databases, it should almost always be the type chosen.

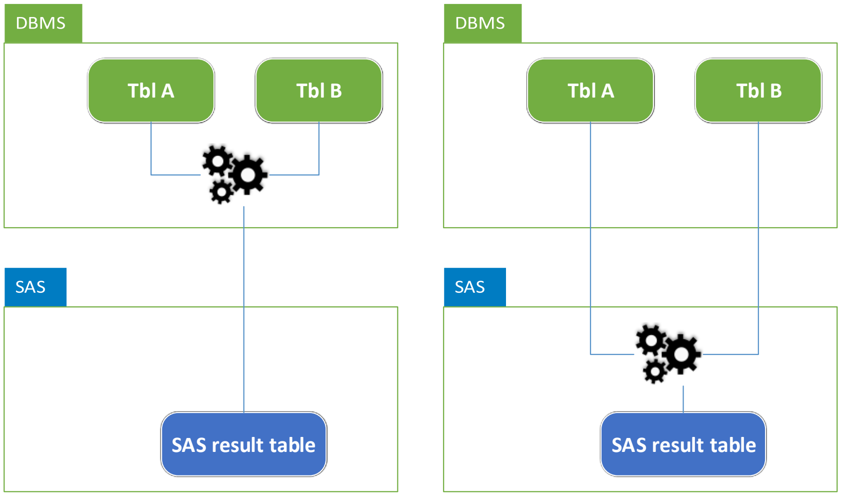
- By metadata library engine

  This type is used to manage security access at the metadata level. For SAS® base libraries, it is an obvious choice but when it comes to external libraries, you should be cautious. Managing security from the metadata server for an external database should be seen as a sort of filtering rather than actual security. Best practices dictate that the data security should be applied and managed at the source (within the actual database).

- By external configuration

  This type should be considered when you want to move the configuration of libnames outside of the metadata (ex: an autoexec file). A typical usage is within an environment with multiple servers linked with SAS/Connect®.

At first glance, using the metadata library engine might seems free of any downsides. However it is important to understand how SAS® works with these connections behind the scenes. Security information being present within the metadata for this library engine, in some cases, SAS® will not have any other choice than to handle queries itself instead of passing them to the database. In these cases, DBMS indexes that would be normally used for the join can't be used. This translates to more data sent over the network, more work space needed on SAS®, and more time required by both the DBMS and SAS® engine to process the query.



SQL Pass through

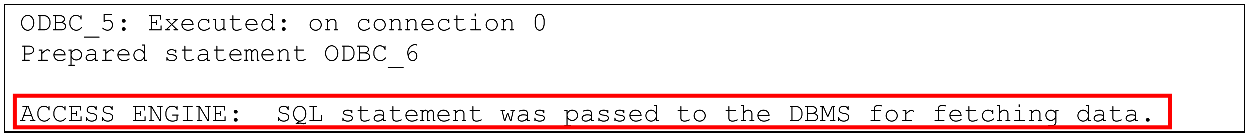
SAS driven join SQL

Figure 3. Representation of a pass through query to a DBMS

Note that using the metadata library engine does not prevent all forms of pass through processing but rather limits its range of action. The metadata library engine really is a nice feature, but if you plan on using it, be aware of its limitations when it comes to mapping external databases.

Testing is definitely your best option in order to ensure proper behavior when it comes to SAS® connecting to your DBMS. One very useful tool is the option "OPTIONS SASTRACE=',,,d' SASTRACELOC=SASLOG NOSTSUFFIX;". It will provide detailed handling of the request in the SAS® log. You'll find below an example of the output provided by that option

```
ODBC: AUTOCOMMIT turned ON for connection id 0

ODBC_1: Prepared: on connection 0
SELECT * FROM "SCHEMA"."TABLE_A" FOR READ ONLY

ODBC: COMMIT performed on connection 0.

ODBC_2: Prepared: on connection 0
SELECT * FROM "SCHEMA"."TABLE_B" FOR READ ONLY

ODBC: COMMIT performed on connection 0.
…

ODBC_5: Prepared: on connection 0
 select ...
 from ...
 where ... FOR READ ONLY

ODBC: COMMIT performed on connection 0.
```



```
ODBC_5: Executed: on connection 0
Prepared statement ODBC_6

ACCESS ENGINE:  SQL statement was passed to the DBMS for fetching data.
```

Output 1. Log output sample generated by SASTRACE

REDUCING THE NUMBER OF CONNECTIONS

The following three tabs in the advanced options panel are “Input”, “Output” and “Input/Output”. They contains options with regards to interaction with the data from the external database. For instance, the input tab contains options to rename columns with names that would be invalid in a SAS® table. The output tab contains options linked to the alteration of data in the DBMS (ex: inserts, updates, deletes). These three tabs should be looked at in detail but in terms of general performance, the following tab is much more important.

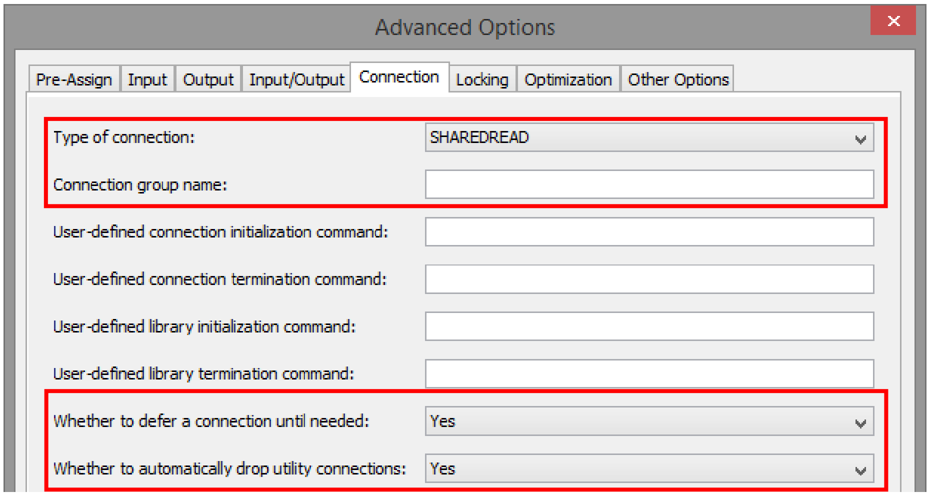
The “Connection” tab contains valuable options that will help control the number of active connections to your external database. With the default options, SAS® will establish one connection per library to the external database plus in some cases utility connections used in queries. These connections add up quickly. The number of sessions is important to limit because most database administrators will fix a limit of concurrent connections. Upon that limit reached, no more connections are accepted. Since some users will start sessions only to interact with SAS® tables, it is needless to have these connections established for them.

One efficient way of controlling the number of sessions is to use the “defer” option. That option will cause libraries to be connected only once they are used a first time. This way, if a library is not used during a SAS® session, it will not get allocated therefore saving a useless connection to the database.

Utility connections are special additional connections that are used by some specific DBMS such as DB2 and Netezza. Each such library can spawn utility connections of its own. Once open, these connections are not released until the SAS® session is closed. Fortunately, the advanced options for these databases allow these sessions to be closed once the connection is no longer required.

The default type of connection is SHAREREAD. In order to limit the number of active connections to the DBMS, this option is a plus. It instruct SAS® to use the same single connection for all read access to the database. This result can also be achieved through the use of connection groups. The option “Connection group name” allows the administrator to group different libraries together by putting the same label in that field. Its use is not required with metadata defined libraries. However, using such group can help easily identify libraries connecting to a common database.

Note that in write/alter cases, a separate connection would be created for the task.



Type of connection: SHAREDREAD

Connection group name:

User-defined connection initialization command:

User-defined connection termination command:

User-defined library initialization command:

User-defined library termination command:

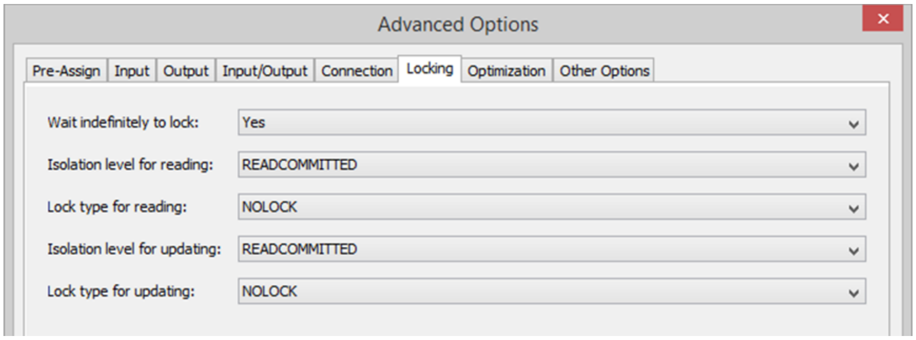
Whether to defer a connection until needed: Yes

Whether to automatically drop utility connections: Yes

Figure 4. Connection options

AVOIDING LOCKS ON DBMS

Another important aspect is the type of access performed on databases. Although not a performance subject, it is an important item to take into consideration. From experience, the default access will work well for Oracle connections. However, when it comes to DB2, the isolation level for reading should be set to “Uncommitted read” or UR to avoid lock on tables. For other database engines, you should consult your database administrator for advice on which options to set.



Wait indefinitely to lock: Yes

Isolation level for reading: READCOMMITTED

Lock type for reading: NOLOCK

Isolation level for updating: READCOMMITTED

Lock type for updating: NOLOCK

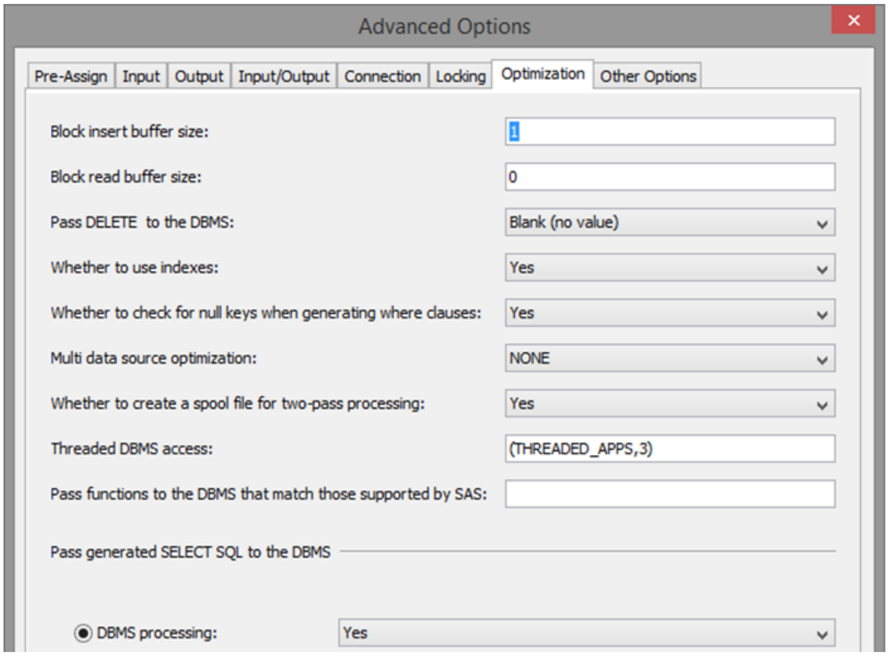
Figure 5. Locking options

OPTIMIZING YOUR CONNECTIONS

The last important tab is the Optimization one. These options will not especially control the number of connections to the database but rather how these are used. Amongst other things, they allow you to tweak the read/write buffers.

Writing to external database is a touchy subject. Whenever such a task is to be performed, it is good practice to establish manually the library with the appropriate options. Then use that connection for the insert and finally clear it once completed. This allow custom performance tweaks that would not be possible with a static common library. If you disagree then this tab will be particularly important to you.

So if you plan on tweaking these options, you should definitely consult your database administrator. Some of these options can have hazardous effects so you should not change a default setting without knowing exactly what it affects.



Block insert buffer size: 1

Block read buffer size: 0

Pass DELETE to the DBMS: Blank (no value)

Whether to use indexes: Yes

Whether to check for null keys when generating where clauses: Yes

Multi data source optimization: NONE

Whether to create a spool file for two-pass processing: Yes

Threaded DBMS access: (THREADED_APPS,3)

Pass functions to the DBMS that match those supported by SAS:

Pass generated SELECT SQL to the DBMS

● DBMS processing: Yes

Figure 6. Optimization tab

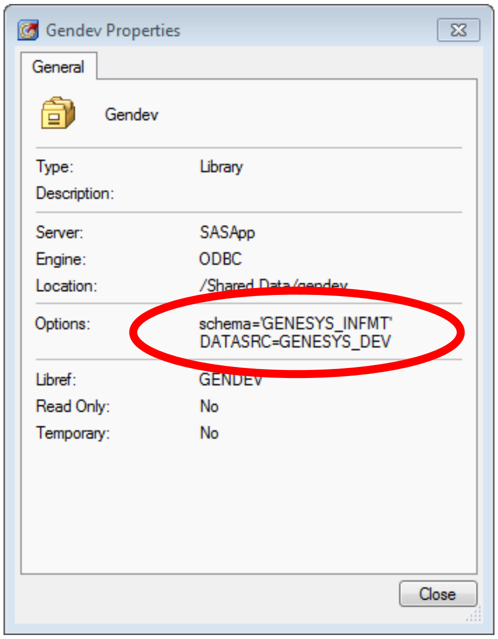
PUTTING IT ALL TOGETHER

A lot has been covered so far in order to limit the number of active connections to external databases. For single library configuration, these tricks will be really efficient. However, when it comes to connecting multiple schemas from a single remote DBMS to libraries, different challenges arise. You want to ensure that pass through queries will be sent to the DBMS even when using different libraries.

To ensure that behavior, all of the libraries connected to a database must have the same options and be connected to the same database server. A slight difference (such as forgetting to specify uncommitted read) between libraries can make SAS® handle the join instead of pushing it to the DBMS.

To guarantee that your library definitions are the same for a given database, the recipe for success is to establish and maintain guidelines for the definition of these libraries. Each database should have its own set of options. The use of guidelines also provides documentation on the configuration.

Another way is to display the library statement for these libraries within SAS/Enterprise Guide® and compare them. This can be achieved by accessing the properties of the library (right click menu). As you can see in the example below, the Options section displays the detailed options that are used to connect.



Gendev

Type: Library
Description:

Server: SASApp
Engine: ODBC
Location: /Shared Data/gendev

Options: schema='GENESYS_INFMT' DATASRC=GENESYS_DEV

Libref: GENDEV
Read Only: No
Temporary: No

Figure 7. Library properties window in SAS/Enterprise Guide®.

CONCLUSION

Connecting external databases is no trivial task. If done without the proper care, the impacts can be severe in terms of performance. With deep knowledge of the library options, setting up guidelines or best practices is the first logical step. Much like base SAS® programming, it is important not to go into low level details as opinions will vary from one expert to another. We must really establish the high level guidelines and dive deeper only when required.

The first element has to be constancy. Make sure that you use the same recipe when it comes to creating your libraries. Make your life easier, several libraries pointing to schemas from the same database should have a similar names (if possible) such as the same two character prefix so that end users know what they are dealing with. Ensure that you use native library engines for pre-assigned libraries and that you do so for all of your libraries connected to a same DBMS. Make that Integrity extend to the individual options you select for libraries connected to a single external database.

Secondly, adjust all possible options (such as the defer option) so that you limit the number of connections and the overall time that they are actives. Informational datamarts are often shared with other BI or reporting softwares. Impacting other SAS® users because of a lockdown (due to the number of connections to the database) is already problematic, the situation gets even worst when users that are not even using SAS® gets impacted.

Finally, use your DBMS specific options. Do take the time to experiment if needed. This time will be well spent if it allows less impacts on the databases that you are using.

REFERENCES

Nelson, Gregory. “Pre-assign SAS libraries? If so, which method?” May 15, 2013. Available at http://blogs.sas.com/content/sgf/2013/05/15/pre-assign-sas-libraries-if-so-which-method/.

SAS® Institutes. “Accessing DB2 Data with SAS9.” Available at DB2 connection information: http://www.sas.com/partners/directory/ibm/db2_sas9.pdf.

SAS® Institutes. “SAS(R) 9.4 Intelligence Platform: Data Administration Guide, Fifth Edition.” Available at http://support.sas.com/documentation/cdl/en/bidsag/68193/HTML/default/viewer.htm#p1icta556r8bacn1ns9kvvnxd4iv.htm.

SAS® Institutes. “SAS/ACCESS(R) 9.4 Interface to PC Files: Reference, Third Edition.” Available at http://support.sas.com/documentation/cdl/en/acpcref/67382/HTML/default/viewer.htm#n0yx5o6hz6nts6n0zi844i9fsc2l.htm.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Mathieu Gaouette
Videotron s.e.n.c.
mathieu.gaouette@videotron.com