Technical Paper
Angela Hall

Using SQL Pass Thru to Improve Web Report Studio Performance

Modifying Information Maps to utilize the highest performing technology

Table of Contents

Introduction

Accessing Oracle, SQL, or other relational data stores using SQL Pass-Thru functionality is significantly more efficient than standard SAS® SQL steps. To get this functionality in SAS® Web Report Studio, custom Stored Processes are required to complete the SQL pass thru. This presentation includes instructions on how to implement this performance improving mechanism on your SAS® BI Server architecture.

SQL Pass Through vs Libname Statements

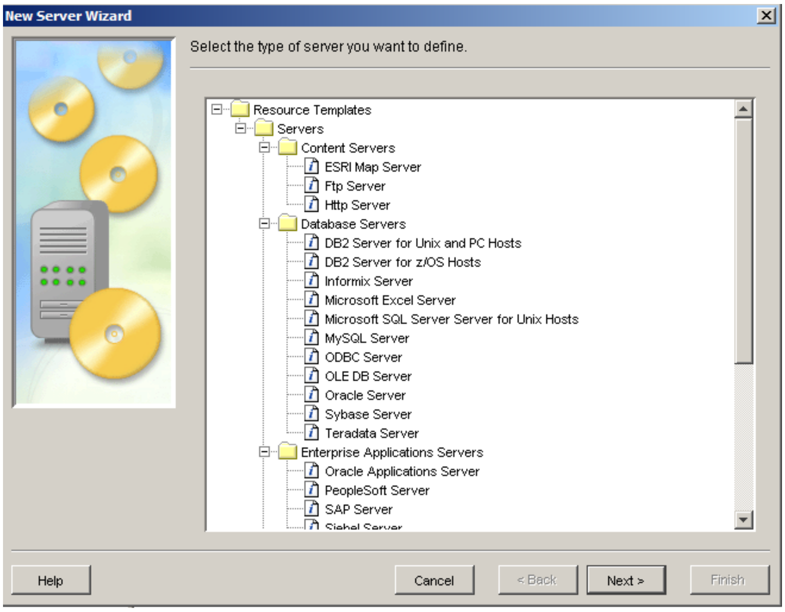
The SQL Pass-Thru is explicit and submits the desired SQL directly to the data source system. Implicit access, via the libname reference, essentially runs SQL code against a SAS WORK table copied from the data source system. Therefore, when querying an Oracle or Teradata table with 50 million records it is clear that the explicit pass thru is more efficient as it sends the query to the data source system and returns to SAS WORK only the matching data records.

Running a test of these two mechanisms on your system is simple and can help qualify the needed development time to migrate to the SQL Pass Thru functionality in SAS Information Maps. Reference material is included at the end of the paper to provide results from other SAS customers of the same test.

Standard Information Map Development

For many SAS customers, the standard development route is taken in SAS Information Map.

1. Define the Oracle/Teradata/SQL Server and Schema in SAS Management Console Servers Node



Select the type of server you want to define.

2. Define the tables for the source system in a SAS Management Console Library

3. Utilize the defined table in SAS Information Maps

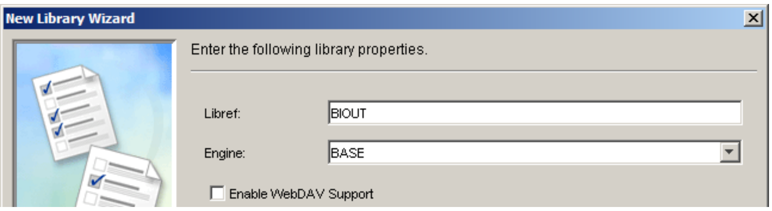
However when linking to a defined external data source in the SAS Metadata from SAS Information Maps, the information map ONLY uses the libname mechanism. This can create significant performance delays in SAS Web Report Studio.

High Performing Information Maps

The only way to improve SQL performance is to create a SQL pass-thru query in SAS, save as a Stored Process, and link the stored process to a SAS Information Map. Included are the steps to perform this linkage.

Steps

1. Create a BASE SAS Library Location in SAS Management Console to place query results METADATA. This physical path will never contain the actual query results of SAS Web Report Studio Users. This table will only reside in this location and serve as a metadata map for the results.



Enter the following library properties.

Libref: BIOUT

Engine: BASE

☐ Enable WebDAV Support

   a. The table libref defined here must be the same as the libref name in the SAS Stored Process Created in Step 4 below.

2. Create a sample query result table using SAS and store into the previously defined BASE SAS Library location

   a. This table can have 0 records. It is only used to aid in defining the metadata correctly in step 3 below.

```
libname biout 'c:\projects\oracle\queryresults';
proc sql feedback  stimer;
connect to oracle (user=xxx password=xxx1 path="@GEMoney" preserve);

create table biout.imap1 as
select *
from connection to oracle
          (select  ... );

disconnect to oracle;
quit;
```

3. Import the Base SAS table into Metadata (via SAS Management Console or a Proc Metalib statement)

4. Modify the sample table creation program above as mentioned below:

   a. Must have *ProcessBody; in the top of the program (This is created by Enterprise Guide automatically, but if creating the Stored Process in BASE SAS, this must be added by the user in addition to the %stpbegin; and %stpend; statements.)

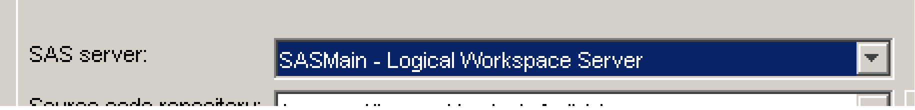
   b. Must reassign the library location libref to libname biout (WORK);

c. **Must be the same libref name of the SAS Management Console Library in step 1 above.**

d. Add Stored Process parameters for SAS Web Report Studio users to enter for the query. References are provided below to assist with coding parameters in Stored Processes.

5. Define the program as a Stored Process in SAS Metadata (via SAS Management Console or SAS Enterprise Guide)

6. The SAS Stored Process must be defined to the Logical Workspace Server. Otherwise it will not appear available when adding it to the Information Map in step 9 below.



SAS server: SASMain - Logical Workspace Server

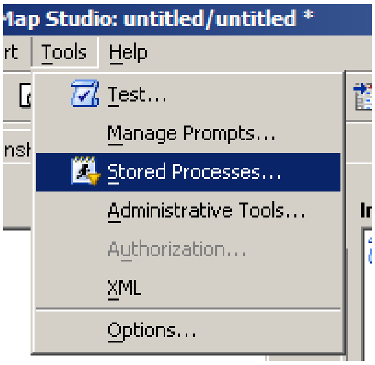
7. Assign user parameters for query

   a. Parameters for user's SAS Web Report Studio prompts MUST be defined at the SAS Stored Process rather than within the Information Map. This is to ensure that the full source table is queried within Oracle/Teradata/SQL/etc first before bringing into the SAS WORK location.

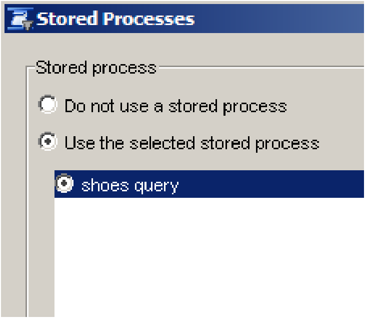
8. Start SAS Information Map Studio and include the sample table (with 0 records) created and defined in steps 1-4 above.

9. Include the SAS Stored Process created in Step 4 and 5 above in the SAS Information Map

   a. Use the Tools -> Stored Processes Screen



   b. Select the query Stored Process defined in Step 5.



Stored Processes

Stored process

○ Do not use a stored process

● Use the selected stored process

shoes query

10. Run an Information Map Test

    a. Select ‘View Query’.

    b. The BIOUT library matches the SAS Management Console Library Definition. This is automatically created by SAS Information Map Studio and helps clarify why the statement to assign the libname to work (libname biout (WORK);) was done in the Stored Process. The output of any query doesn’t need to be saved or managed in a physical location on the server.

```
Options Locale=en_US;
LIBNAME BIOUT BASE 'c:\projects\oracle\queryresults';
```

    c. Note that the Stored Process execution is completed prior to the statement to create the test table.

```
StoredProcess+omi://Foundation/reposname=Foundation/ClassifierM
ap;id=ASHMEEDH.B1000R1D
```

    d. Also note that the Proc SQL creation statement now matches what was selected in the Information Map Test GUI screen.

```
Proc SQL; Create Table %DATA% as SELECT
        <test selection>
FROM
        BIOUT.imap1
;
quit;
```

    e. Close the Query View Screen and select 'Run Test’. The Stored Process Parameters defined in Step 5 will appear for the tester.

11. Utilize this map in SAS Web Report Studio to create a report.

Conclusion

Data administrators will need to shift prompt creation from the Information Map to the Stored Process however the performance improvements of utilizing SQL Pass-Thru will be significant and immediately apparent to the user community accessing these sources via SAS Web Report Studio.

References or Resources

How to Create a SAS® Stored Process and Then Associate It with a SAS® Information Map, **http://support.sas.com/kb/26/175.html**

How to Dynamically Select a Table for a SAS® Information Map Using a SAS® Stored Process with Prompts, **http://support.sas.com/kb/33/146.html**

Going Beyond Simple Information Maps to Improve Access to Data Sources, **http://www2.sas.com/proceedings/forum2008/049-2008.pdf**

The Libname Engine Compared to SQL Pass-Through, **http://www.nesug.org/proceedings/nesug07/po/po04.pdf**

Parameter Coding and Reserved Macros

http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/input.html

http://support.sas.com/rnd/itech/doc9/dev_guide/stprocess/reserved.html

Special Thanks

Penny Tinkler, Senior Quantitative Analyst for GE Consumer Finance-Americas, assisted with testing and verifying the above scenario on an Oracle dB.