Quick Tips: Automatically Submitting SAS Code When Server is Connected

Intro

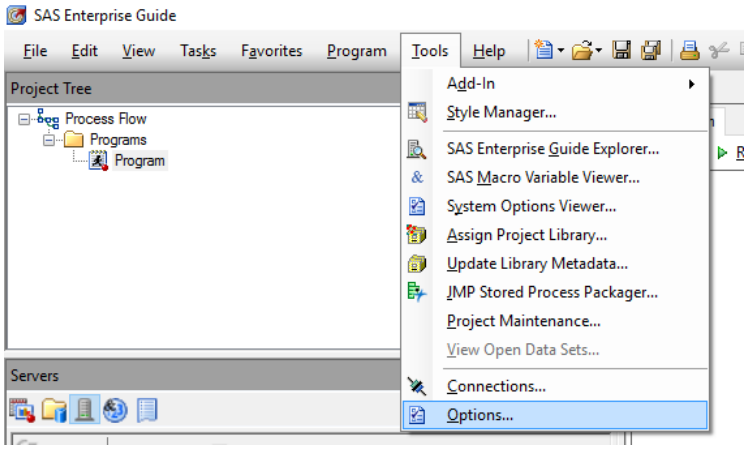
This quick tip may be of interest to those in the SAS user community using SAS-EG with server set-up. Those who would like the ability to customize the SAS-EG environment by having a block of code submitted immediately whenever they launch SAS and connect to a server may find it is easy to implement and make their experience using SAS more efficient.

Use Cases

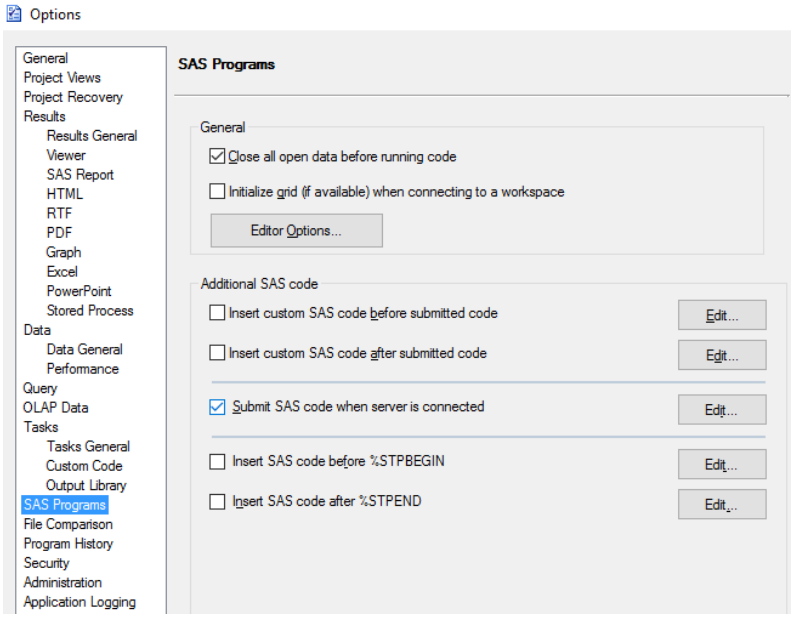
There are likely many small tasks you perform regularly, but with minor periodic updates that make it undesirable to create separate stored procedures for. Think of anything you want the agility to change with some regular frequency, yet make applicable to all your projects and codes at the same time. You write it once, test it out, and maintain in one place -- then have it consistently applied at the start of every session. The first thing that comes to my mind is Libnames -- I'm always resetting passwords!

Steps to set-up

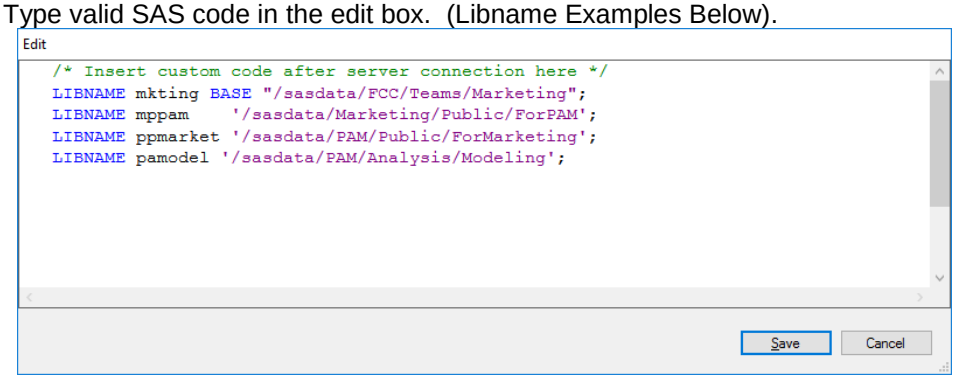
1. In Enterprise Guide, Click Tools, Options…



2. Click SAS Programs, Check Submit SAS code when server is connected, click Edit…



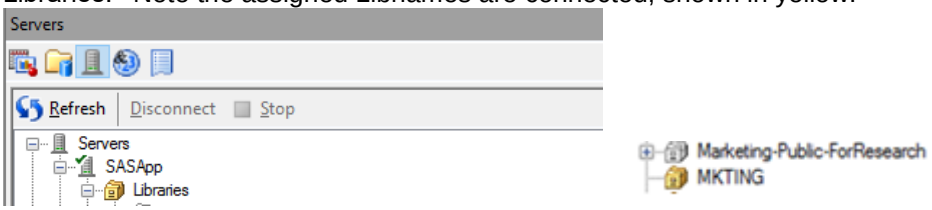
3. Type valid SAS code in the edit box. (Libname Examples Below).



4. Also, save a copy of this code elsewhere. Trust me.

5. Click Save. Close and then restart EG.

6. After restarting SAS-EG, navigate to where your Libname points. In my case, 'Servers, SASApp, Libraries.' Note the assigned Libnames are connected, shown in yellow.



Note: I've found standardizing Libname aliases great for encouraging collaboration amongst those you work with; which makes it easier for folks to share SAS codes or projects.

Hey! Question? Where did the Custom Code you just typed get Saved?

- The code will be saved in some default location (C:Drive) set at installation
- Work of Caution: This gets deleted/overwritten when an upgrade or new version is pushed.
- *Warning: Save a back-up copy! (Copy and paste it into an Email to yourself, OneNote, etc.)*

Some Common Applications to Consider

- Libname connections
- Password management
- Declare macros

/*==========*/

Let's take this to the next level !!

/*==========*/

SQL Pass-thru in the SAS-EG Server Environment:

An Example using Libnames and Macros, for Managing Credentials and Connections

Intro

This paper shows a practical example of implementing credential and connection management using Libnames and Macros. In short, we write SAS code that will automatically run when your SAS-EG session is launched, and your user session connects to the server.

Examples following will show how this can be used to connect to data on an Oracle server, via SQL Pass-through. Keep in mind that similar connections can be made to other environments.

Use Case: Benefits of SQL Pass-thru

Let's say you have data in various systems residing on numerous server locations. Some of the datasets are large, and when you connect to them with a Libname connection you find the performance to be slow.

Rather than pulling the large data files whole, into the SAS environment, before applying filters, you would prefer to just return the result-set across the network to your SAS Server environment.

Note: Remember to treat your DBA nicely, as you may have to work together to test out your connection path strings, given the settings

Example Code: Libnames and Macros, for Managing Credentials and Connections

Step 1: Declare the Libnames you will require (Oracle).

/* Declare Libnames */

```
Libname SAP3  ORACLE PATH='ask_dba'  SCHEMA= 'ask_dba'   USER='yours'   PASSWORD= 'yours';
```

Step 2: Declare any macros or alias required

/* Declare Macros */

```
%let strDbSAP = 'your_path';       %let strUser = 'your_user';      %let sapPwd = 'your_password';
```

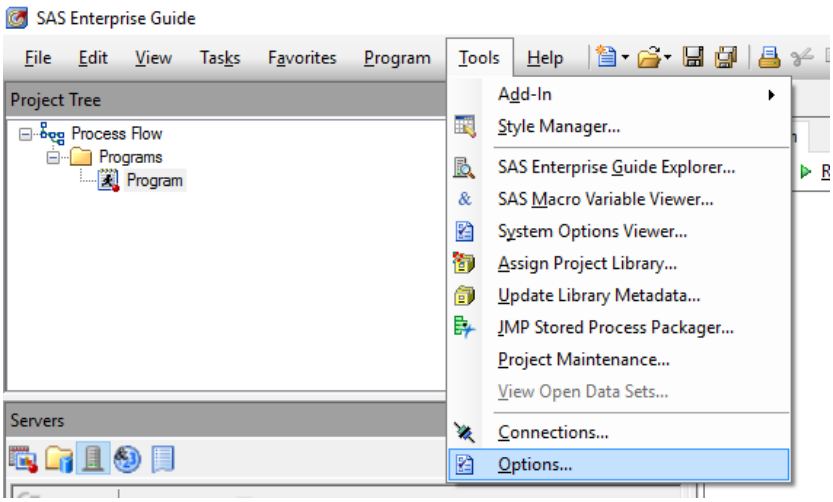
Example Query: SQL Pass-Through, Connection to Oracle

Step 3: Write SQL code, wrapped in a connection to oracle syntax with required credentials

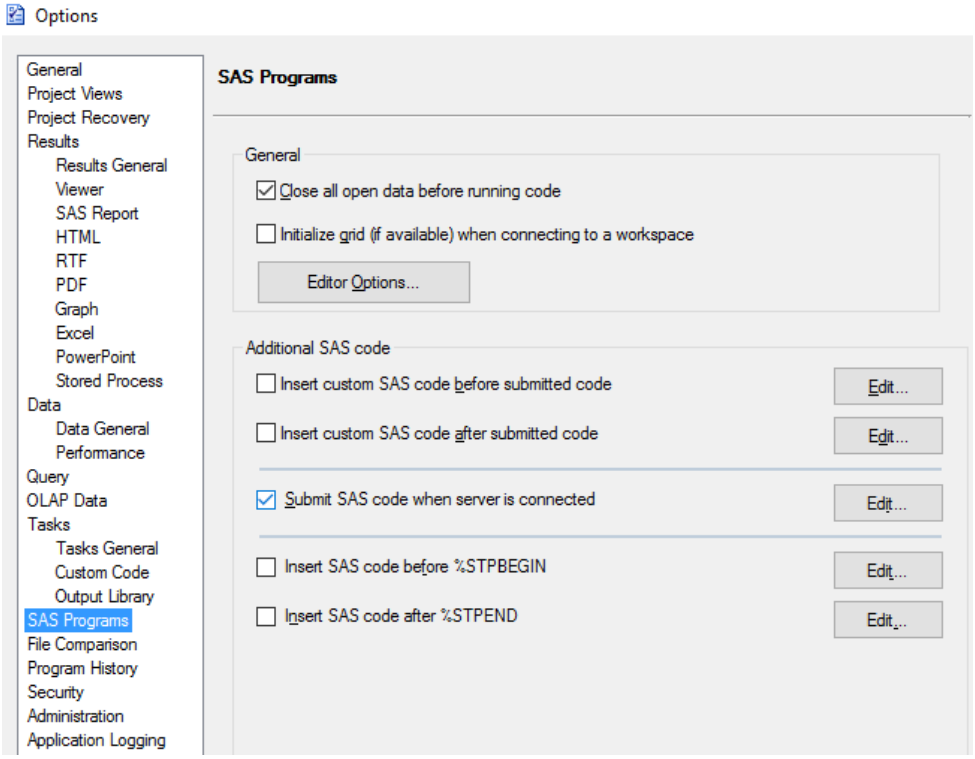
```
proc sql;
 connect to oracle (path=&strDbSAP user=&strUser password=&sapPwd);

    create table work.vzzkoko_delfz_t1 as select * from connection to oracle (

        SELECT to_date(delfz, 'yyyymmdd') as delfz, rkey, pkey
        FROM sap3.vzzkoko
        WHERE substr(delfz,5,2) in('01','02','03')   );
 disconnect from Oracle;
quit;
```

Managing Credentials and Connections: Have them run whenever you connect to SAS server

1. In Enterprise Guide, Click Tools, Options…



2. Click SAS Programs, Check Submit SAS code when server is connected, click Edit…



3. Type your valid SAS code in the edit box. Save a copy elsewhere, and save when done.
   /* Declared Libnames from Step 1*/
   /* Declared Macros from Step 2*/

4. When next you launch SAS-EG, the connection to the server will automatically execute the code blocks we've saved.
5. To complete this example, run **SQL Pass-Through, Connection to Oracle example.**
   *** Note: See other options to insert custom SAS code before or after submitted code ***