

# Passwords and AuthDomain

Debby Gear

December 9, 2016

DG Consulting

# Passwords

- `libname myDB odbc database=ADB user=dgear  
using=MyPassword;`

# Using a Hidden File

- Use protected file that cannot be read by other people to store our credentials.
- pwd.sas
- Contents
- %let userid=dgear;
- %let pwd= OOPSmyPWD;

# Using a Hidden File

- `%include '~/pwd.sas';`
- `libname myDB odbc database=ADB user="&userid."  
using="&pwd.";`

# Using a Hidden File

- Works great UNTIL
- Options symbolgen;
- SYMBOLGEN: Macro variable USERID resolves to dgear
- SYMBOLGEN: Macro variable PWD resolves to OOPSmypwd
- `database=ADB user="&userid." using="&pwd.";`
- NOTE: Libref MYDB was successfully assigned as follows:
  - Engine: ODBC
  - Physical Name: ADB

Alternative

**There must be a better way**



# AuthDomain

## AuthDomain

Your SAS administrator creates an authentication domain definitions while creating a user definition with the User Manager in SAS Management Console. The authentication domain is associated with one or more login metadata objects, which provide access to the server. The authentication domain is resolved when the DBMS engine calls the SAS Metadata Server and returns the authentication credentials.

# AuthDomain

## AuthDomain

The AUTHDOMAIN= option enables you to avoid explicitly specifying a database user ID and password in a SAS program. You define an authorization domain in metadata and define the associated credentials in a metadata repository. Then, you can simply reference the authorization domain in a LIBNAME statement or in the SQL procedure pass-through code.

# AuthDomain

- The SAS/ACCESS engine accepts the credentials as if the user specified them in the USER= and PASSWORD= options.
- The authentication domain and the associated login definition must be stored in a metadata repository and the metadata server must be running in order to resolve the metadata object specification.

sasv9.cfg

- Modifying the config file
- sasv9.cfg located in your home directory

sasv9.cfg

```
metauser="metadata-userid"  
metapass="metadata-password"  
metaport=8561  
metaprotocol=bridge  
metarepository="metadata-repository"  
metaserver="server-name";
```

# Connect with AuthDomain

- `libname myDB ODBC database=ADB authdomain='ADB_Auth';`
- NOTE: Libref MYDB was successfully assigned as follows:
  - Engine: ODBC
  - Physical Name: ADB

# ODBC.INI

- How this is done
- Driver definitions are defined in the odbc.ini file

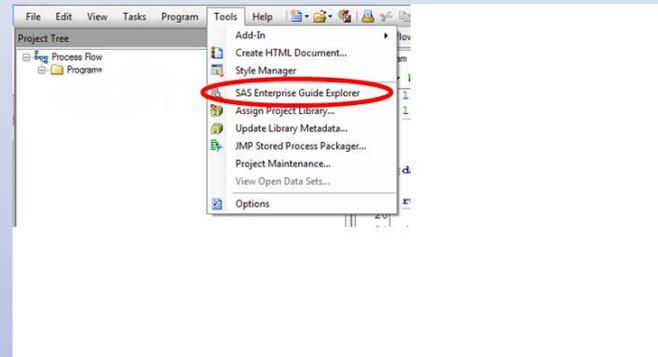
- Contents

ADB\_Auth = NetezzaSQL

- In addition, a metadata file is created which contains the user credentials and will be used for authenticating logins.

# Define the Connection

- The SAS User can create the connection using SAS EG
- Select -> Tool, SAS Enterprise Guide Explorer



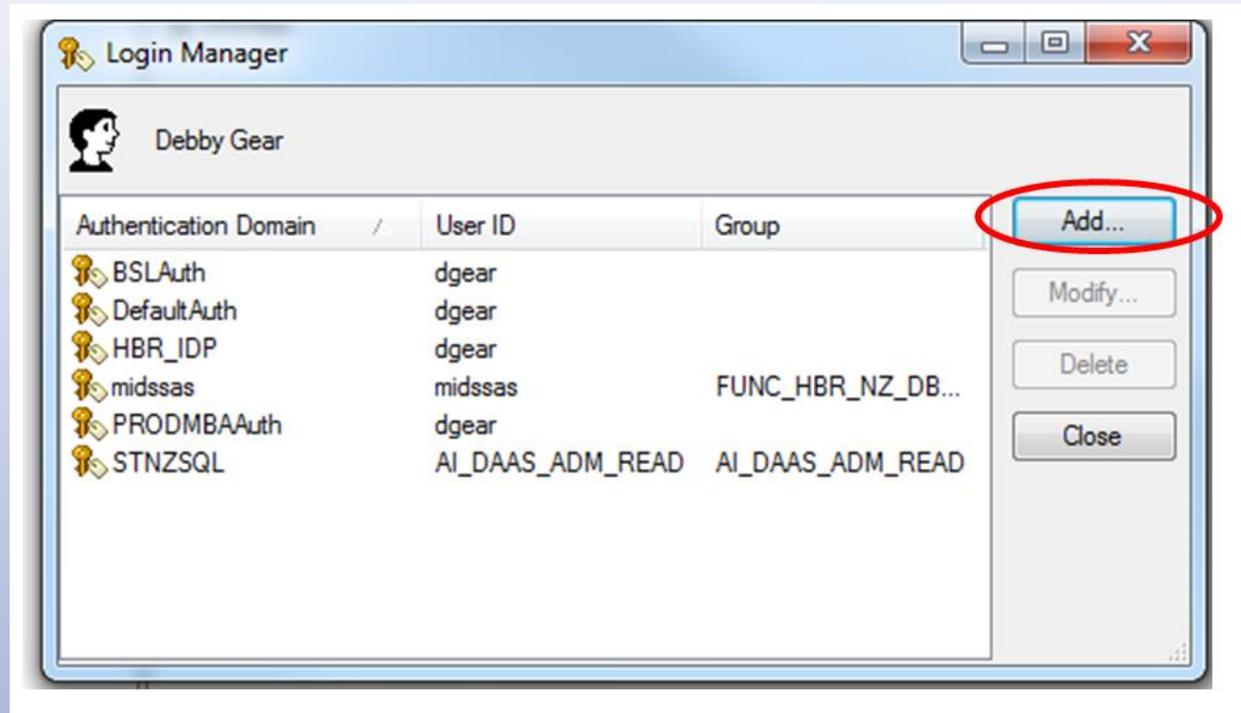
# Define the Connection

- Select Manage Logins



# Define the Connection

- Select Add and enter credentials, select ADB\_Auth from the drop down box.



Questions



# Thanks

- <http://support.sas.com/documentation/cdl/en/acredb/69580/HTML/default/viewer.htm#noaiq25zc8u8u6n1i81myoa24sd3.htm>
- <http://support.sas.com/kb/38/204.html>