Paper 3651-2019

Using a File Data Source Name to Centralize Data Connection Management in DataFlux® Data Management Studio

André Harper, United States Census Bureau; Marc Price, Tom Gaughan, Matt Hall, and Michael Bretz, SAS Institute Inc.

ABSTRACT

The United States Census Bureau (USCB) Demographic Systems Division (DSD, collectively USCB-DSD) utilizes Dataflux® Data Management Studio and DataFlux® Data Management Server for data management and cleansing. USCB-DSD accesses DataFlux Data Management Studio via a Microsoft Windows virtual desktop (vDESK) where users can perform data management tasks unique to their individual program area. Due to USCB security restrictions, users do not have access to the local Windows program files where the DataFlux Data Management Studio configuration is stored by default. Users also do not have the capability to establish a System Data Source Name (DSN) as required to make the connection to the repository. To solve these problems, USCB-DSD implemented a shared repository consisting of an Oracle database management system (DBMS) with the DataFlux data connection configuration saved on a shared file system and user credentials saved in the SAS metadata. This paper discusses how the USCB uses DataFlux Data Management Studio, the architecture of the USCB environment, the administrative problems encountered due to security restrictions, and a solution that minimizes the administrative problems.

INTRODUCTION

The Dataflux Data Management Studio software requires the use of a data repository that typically resides in a DBMS. Users of the software can configure a data connection to a DBMS as needed. However, users in the USCB environment lack the capability to make permanent changes to the list of data connections. As a result, the use of Dataflux Data Management Studio in the USCB was expected to result in significant administrative overhead. This paper discusses how SAS administrators and database administrators can configure DataFlux Data Management Studio to work in a similar environment with minimal administrative overhead by presenting the following topics:

- Background about the USCB
- How USCB uses Dataflux Data Management Studio
- The architecture of the USCB environment
- The administrative problems encountered while using Dataflux Data Management Studio in the secure USCB environment
- An alternate configuration that resolves the problem in the secure USCB environment

BACKGROUND

As the federal government's largest statistical agency, the mission of the United States Census Bureau (USCB) is to serve as the nation's leading provider of quality data about its people and economy. As such, USCB is one of the largest of the SAS government customers. Within USCB, the Demographic Systems Division (DSD) is responsible for consolidating IT resources and functionality in the Demographic Directorate in support of numerous surveys and supplements, both reimbursable and internal. *Reimbursable* or sponsored surveys are demographic surveys that are conducted for other government agencies. The reimbursable surveys include the Current Population Survey, the National Health Interview Survey, and the National Survey of College Graduates. *Internal* DSD demographic surveys measure income, poverty, education, health insurance coverage, housing quality, crime victimization, computer usage, and many other subjects. Within the USCB, these include the American Housing Survey (AHS) and the American Community Survey (ACS). As you can imagine, USCB-DSD has a very complex task in managing resources, guaranteeing that security measures are in place, and ensuring that users have proper access to get their work done.

HOW USCB USES DATAFLUX DATA MANAGEMENT STUDIO

Historically and currently, USCB-DSD utilizes Base SAS® programs, either running in batch or in local interactive sessions, to analyze and cleanse data. In 2017, USCB-DSD began a pilot program to test the efficacy of using the SAS® Data Management Standard package for data cleansing, address verification, data profiling, and standardization across the enterprise.

As mentioned, USCB-DSD processes the data for a great number of surveys. These surveys tie directly to the US population, so the size of the data is quite large. In addition, the data arrive from many different sources, thus cleansing and standardization become extraordinarily important as factors for productivity. Fields such as addresses require verification via SAS® Quality Knowledge Base for Contact Information. Also, customizable quality knowledge bases and schemes can be created for a great number of different, common fields across surveys, providing enterprise standardization. For example, the same categorical field (such as Veteran's Status) might have different categories across surveys or even within the same survey across years!

One of the most powerful features of SAS Data Management software is column profiling. *Column profiling* is the act of taking a certain data item within a table and generating a count for each unique value within the data item. From here, you derive frequency distributions and perform further analysis upon the counted values. DataFlux Data Management Studio provides the use of a repository for the storage of these findings. For an individual user of DataFlux Data Management Studio, it is adequate to have a set of files to store the repository contents. However, in a collaborative environment, such as the one that the new USCB-DSD architecture provides, this type of storage is not adequate. Hence, USCB-DSD needed a relational database management system (RDBMS) to serve the function of storing and sharing the repository contents.

SYSTEM ARCHITECTURE

To understand this project, you need to know about the USCB-DSD's system architecture. Figure 1 shows that the USCB-DSD environment is comprised of four virtual machines (VMs) running on RHEL 6.9 and a bare-metal box running on RHEL 7.4

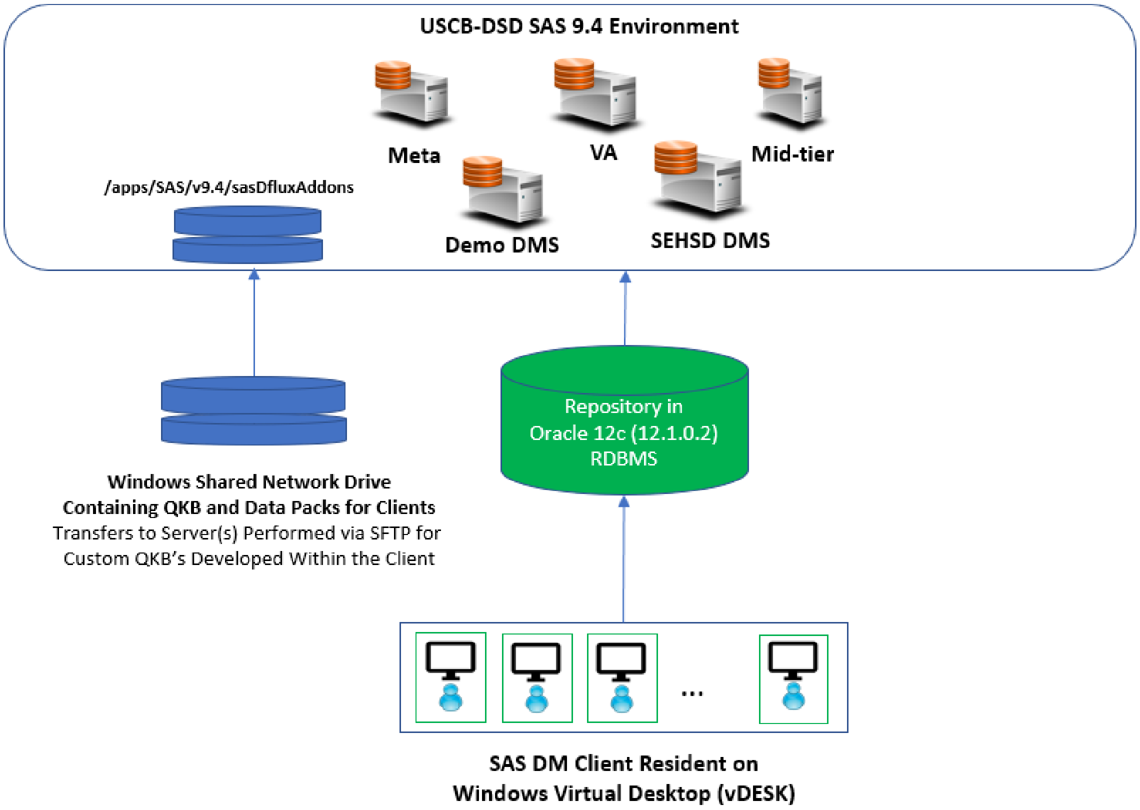

USCB-DSD SAS 9.4 Environment
Meta
VA
Mid-tier
/apps/SAS/v9.4/sasDfluxAddons
Demo DMS
SEHSD DMS
Repository in
Oracle 12c (12.1.0.2)
RDBMS
Windows Shared Network Drive
Containing QKB and Data Packs for Clients
Transfers to Server(s) Performed via SFTP for
Custom QKB's Developed Within the Client
...
SAS DM Client Resident on
Windows Virtual Desktop (vDESK)

Figure 1. USCB-DSD Environment

In Figure 1, the VMs are Meta, VA (SAS® Visual Analytics), Mid-tier, and Demo DMS (the compute tier). The second compute tier, SEHSD DMS, runs on the bare-metal box.

THE ADMINISTRATIVE PROBLEMS

Similar to all US Federal Government agencies, the USCB requires strong security on all IT systems. All servers must have secure connections between data sources and client applications. In the case of the USCB, the privacy and confidentiality of the data sources mainly fall under Titles 13 and 26 of the United States Code. Violation of either title, singly or in combination, carries severe financial and incarceration penalties.

As part of the centralized approach to IT, the USCB provides its employees with a virtualized Windows environment (vDESK). As expected from the above security requirements, there are strong restrictions on what users can and cannot do. These security measures presented a complication for implementing a collaborative SAS Data Management deployment in the enterprise.

The first problem is that anyone outside of the vDESK administration team does not have the local administrative privileges to add a System DSN connection to an RDBMS (Oracle, in this case). This method is standard for creating a DSN for the repository. In addition, when access was granted to the SAS Center of Excellence at USCB, preliminary testing demonstrated that the System DSN settings could not be sustained for the user across vDESK sessions, thereby revealing a very large technical hurdle outside of the control of the interested parties.

The second problem is that there is a need for a shared repository to which each user must have access. DataFlux Data Management Studio uses one of the two following approaches for standard access: providing a set of common database credentials that is shared with all users, or granting the database permissions required to access the common repository to the individual database user-IDs. Neither of these approaches was acceptable due to the following issues:

- First, USCB security policy requires distinct accounts for the repository and does not permit the sharing of credentials with all users.
- Second, even if USCB policy permitted sharing of credentials, it would cause administrative issues. For example, when the shared password changes, mass email coordination would be required along with routing of numerous approval chains through several teams within USCB. Also, any user who forgets to update the password and uses the expired password could potentially lock out other users.
- Third, there are a set of 15 users licensed to use the DataFlux Data Management Studio software at any particular time. This group can change over time. Hence, granting/denying the database permissions to a changing user base is also an administrative issue that would result in many requests to the Oracle database administrators (DBAs).

USCB-DSD needed a solution that allowed the Oracle DBA to do the following:

- Add databases as needed to get relief from the potential administrative issues that were just outlined
- Set up a common repository for centralized data profiles and business rules

THE SOLUTION

After some discussion, USCB-DSD proposed implementing a solution that would centralize the database connection to on-site SAS Federal Consulting services. SAS Technical Support was then contacted to determine whether the high-level idea was feasible and how to implement a solution that would meet USCB security requirements. SAS Technical Support recommended the following two-part approach:

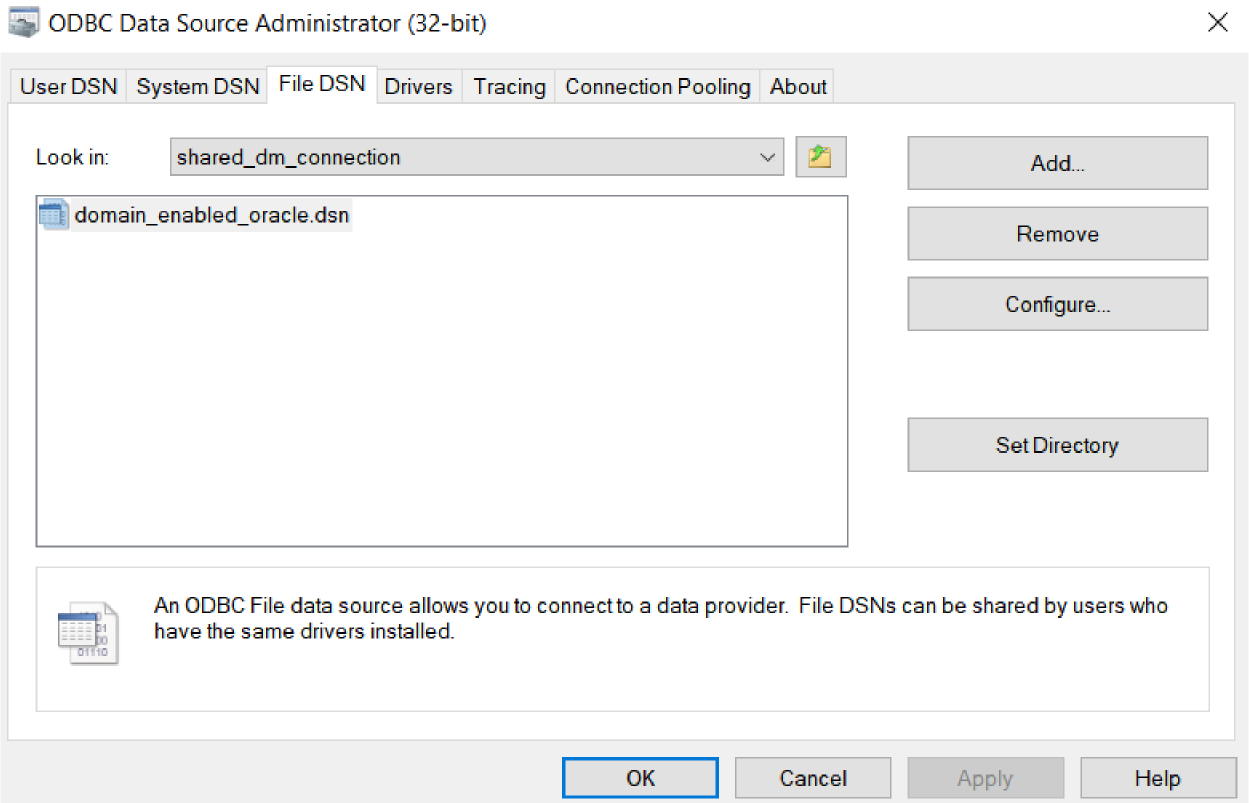
- Use file-based data source names (file DSNs) on a shared network drive.
- Use a domain-enabled ODBC connection that references the file-based DSN.

When USCB-DSD decided to implement this solution, SAS Federal Consulting configured, tested, and validated the environment.

FILE-BASED DSNS

A file-based DSN is a technical term that refers to the use of a flat text file to save the ODBC configuration. However, Microsoft Windows usually saves ODBC connection information in the Windows registry, and users typically use the built-in ODBC Data Sources control panel to configure the ODBC connection. Other operating systems typically use a flat file, often named odbc.ini, to save the ODBC configuration. A little-known fact is that the ability to create a file-based DSN is built into the Windows ODBC Data Sources control panel. The process to create a file-based DSN in Windows is similar to the more familiar process for creating a user or system DSN. However, it has the following differences:

- You perform the configuration on the **File DSN** tab, instead of on the **User DSN** or **System DSN** tabs, as shown in Display 1 below.
- Users are prompted to specify an output location to save the configuration.



Look in: shared_dm_connection

domain_enabled_oracle.dsn

An ODBC File data source allows you to connect to a data provider. File DSNs can be shared by users who have the same drivers installed.

Display 1. The File DSN Tab

DataFlux Data Management Studio requires a one-time configuration change to instruct the software to look in the correct location to use file-based DSNs. The DAC/DSN parameter in the DataFlux Data Management Studio etc\app.cfg configuration file instructs the software to look in the specified location for file-based DSNs. You use the following format on the DAC/DSN parameter to instruct the software to look for file-based DSNs in a network drive:

```
DAC/DSN=\\network_drive\path
```

Note: Replace the italicized text with the relevant information from your environment.

For the USCB-DSD environment, SAS Technical Support recommended using a file-based DSN and saving the file to a shared network drive. The vDESK administrator needs to make this one-time change and save the change to the vDESK image. With this approach, users do not need to configure any data connections. The administrative team can create or update the file DSNs in the network drive at any time, and no further action is required by the users. This approach also resolves the issue of losing DSNs across vDESK sessions.

This one-time configuration of updating the app.cfg configuration file, and then using file-based DSNs saved to a network location resolves the first part of the problem. However, one problem remains: ODBC connections require user credentials. The USCB-DSD was also looking for a way to minimize the administrative overhead of managing database user credentials.

DOMAIN-ENABLED ODBC CONNECTIONS

A domain-enabled ODBC connection is a documented and common way to save credentials in a SAS® 9.4 environment. From a security perspective, this is a far safer method for an environment as concerned with privacy as the USCB. Hiding the connection credentials in the SAS metadata and granting group access also ensures that the only people with access are those that should have access. No authorized user could surreptitiously pass the credentials to a non-authorized user and the knowledge of the credentials is minimized.

You must complete three separate configuration tasks to use domain-enabled ODBC credentials with DataFlux Data Management Studio:

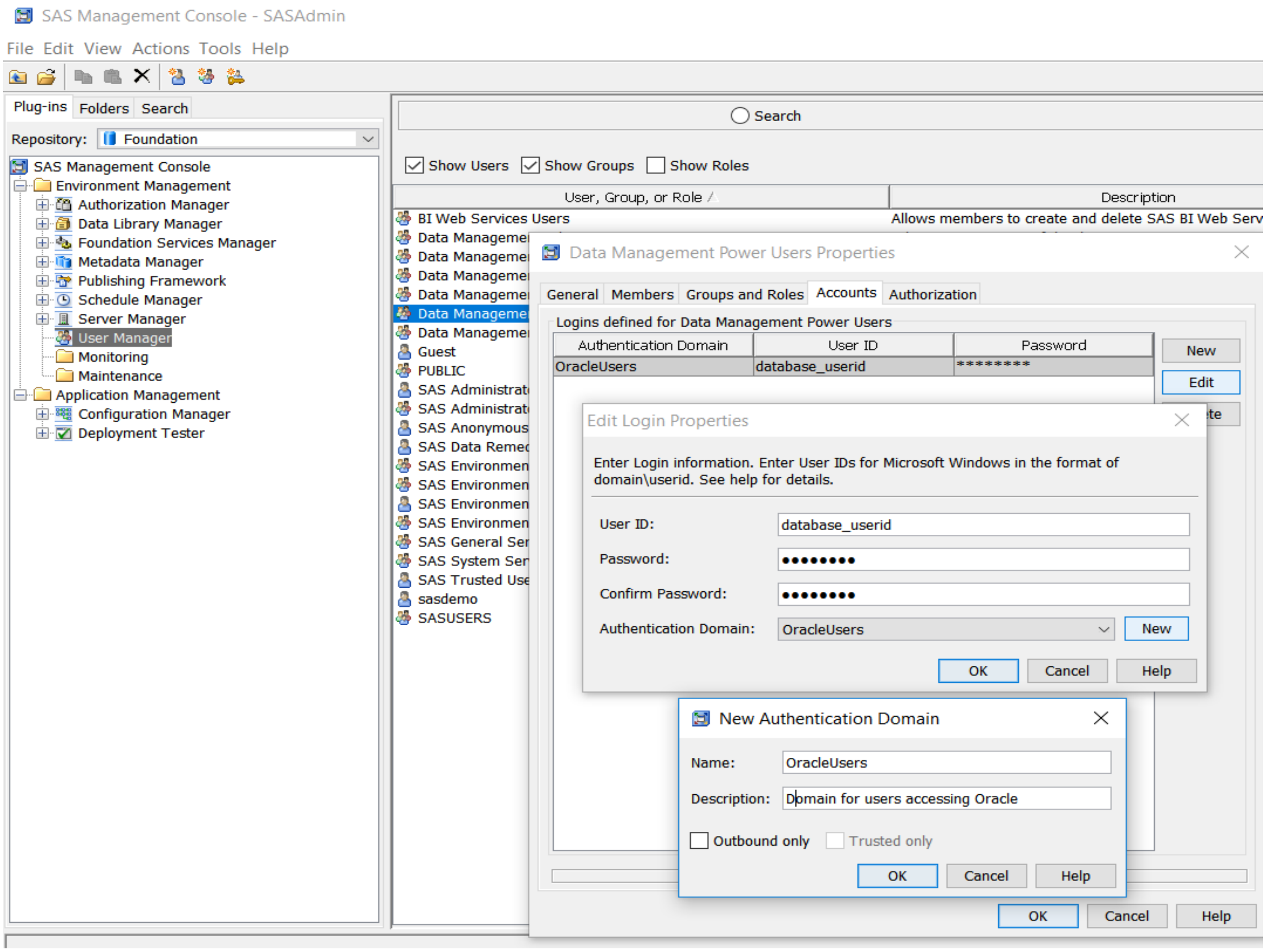
- Save the authentication domain and associated credentials in the SAS metadata.
- Configure DataFlux Data Management Studio to use SAS® Metadata Server for user authentication.
- Configure DataFlux Data Management Studio to use to use the authentication domain.

Task 1: Create an Authentication Domain

To complete the first task, a SAS administrator must follow these steps:

1. Sign into SAS Management Console and connect to the SAS Metadata Server.
2. Navigate to the **User Manager**.
3. Right-click a user or group and select **Properties**.
4. Navigate to the **Accounts** tab and click **New**. The **New Login Properties** dialog window opens.
5. Specify the database user ID, database password, and authentication domain to associate with the database credentials. If needed, click the **New** button next to the **Authentication Domain** field to create an authentication domain.
   Caution: The authentication domain name must not exceed the maximum length (25 characters) allowed by DataFlux Data Management Studio.

Display 2 shows an example in which the administrator added an authentication domain named OracleUsers to the **Data Management Power Users** group in the SAS metadata.



Repository: Foundation

☑ Show Users ☑ Show Groups □ Show Roles

Allows members to create and delete SAS BI Web Serv

Logins defined for Data Management Power Users

Authentication Domain	User ID	Password
OracleUsers	database_userid	********

Enter Login information. Enter User IDs for Microsoft Windows in the format of domain\userid. See help for details.

User ID: database_userid

Password: ••••••••

Confirm Password: ••••••••

Authentication Domain: OracleUsers

Name: OracleUsers

Description: Domain for users accessing Oracle

□ Outbound only □ Trusted only

Display 2. Example of Adding an Authentication Domain

After the administrator creates the authentication domain and adds credentials to the group, any user in that group has access to that set of credentials.

Task 2: Configure DataFlux Data Management Studio to Use SAS Metadata Server for User Authentication

To complete the second task, a SAS administrator must configure DataFlux Data Management Studio to authenticate against the SAS metadata by adding the following two options to the DataFlux Data Management Studio etc\ui.cfg file:

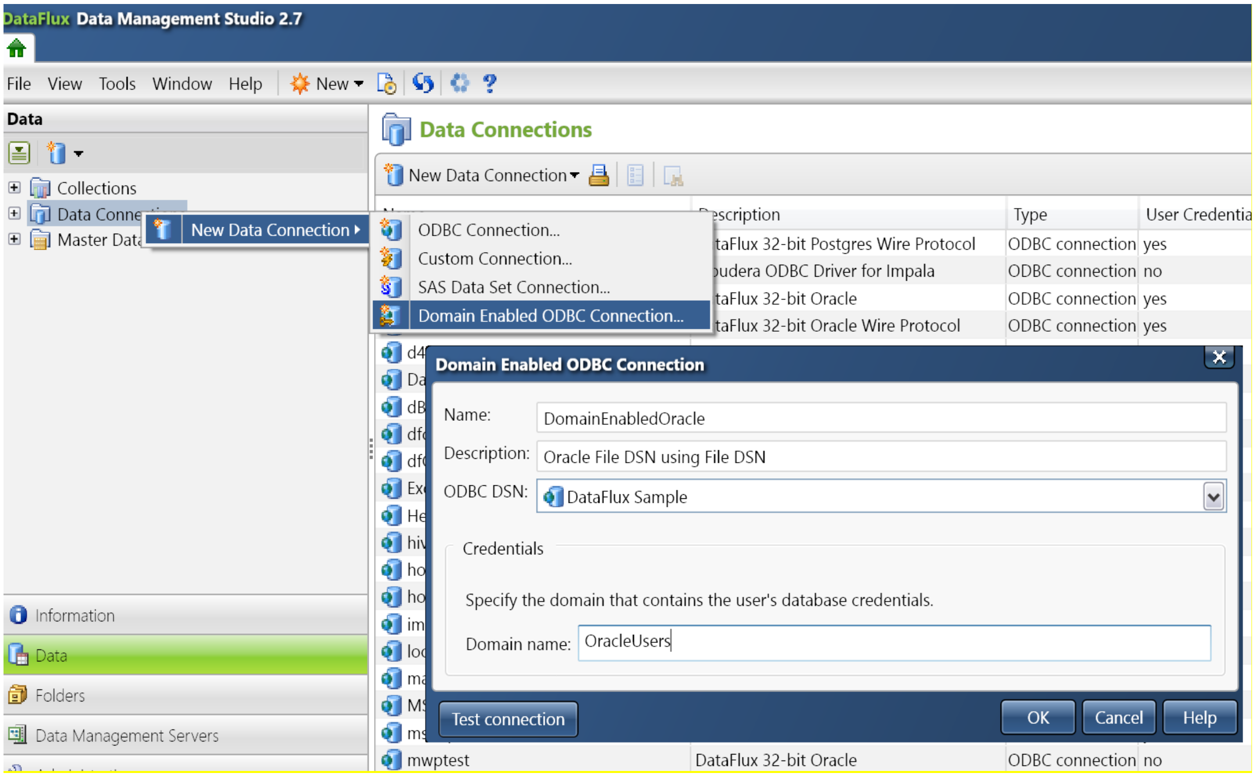
```
BASE/OMR_PROMPT_ON_STARTUP=TRUE
BASE/AUTH_SERVER_LOC=iom://metadataserver.yourcompany.com:8561
```

These two options instruct DataFlux Data Management Studio to prompt for SAS metadata user credentials at startup and verify those credentials with the specified SAS Metadata Server. Setting these options in the etc\ui.cfg file makes both options apply to all users of DataFlux Data Management Studio. For the USCB-DSD secure vDESK environment, the administrator needs to complete this setup only once, and then the settings can be saved to the vDESK environment to become a permanent part of the image.

Task 3: Configure DataFlux Data Management Studio to Use the Authentication Domain

This last configuration step associates the DSN with the authentication data in the SAS metadata. To complete the first task, a SAS administrator must follow these steps:

1. In DataFlux Data Management Studio, navigate to the Data Riser, right-click **Data Connections**, and select **New Data Connection ➔ Domain Enabled ODBC Connection**.
2. In the Domain Enabled ODBC Connection dialog box, the administrator creates a name for the connection, an optional description, and selects which ODBC DSN to use and which authentication domain to use. The administrator must select a regular ODBC DSN instead of the desired file-based DSN because file-based DSNs are not included in the selection list for the ODBC DSN. Display 3 shows an example in which the administrator selected the Dataflux Sample DSN and associated that DSN with the authentication domain.



Name	escription	Type	User Credentia
	taFlux 32-bit Postgres Wire Protocol	ODBC connection	yes
	udera ODBC Driver for Impala	ODBC connection	no
	taFlux 32-bit Oracle	ODBC connection	yes
	taFlux 32-bit Oracle Wire Protocol	ODBC connection	yes
mwptest	DataFlux 32-bit Oracle	ODBC connection	no

Domain Enabled ODBC Connection

Name: DomainEnabledOracle

Description: Oracle File DSN using File DSN

ODBC DSN: DataFlux Sample

Credentials

Specify the domain that contains the user's database credentials.

Domain name: OracleUsers

Display 3. Example of Associating a DSN with an Authentication Domain

The Domain Enabled ODBC Connection dialog box saves the configuration to the location specified by the DAC/DSN parameter. If DAC/DSN was not already set, then the default location is **`C:\Users\windows_userid\AppData\Roaming\DataFlux\DMStudio\studio1\etc\dftkdsn`**.

3. Edit this file to correct the DSN. The configuration file has a file name that matches what the administrator specified in the Domain-enabled ODBC Connection user interface and has a .dftk file extension. The configuration file contains XML text that can be edited with any text editor.

4. Edit the **`<attribute>`** tag near the end of the file so that the name parameter is set to "**`FILEDSN`**" and the value of the attribute is set to the file name of the file DSN that was saved to a network location. In the following example, the file might contain the following attribute tag before changes are made:

   ```
   <attribute name="DSN">Dataflux Sample</attribute>
   ```

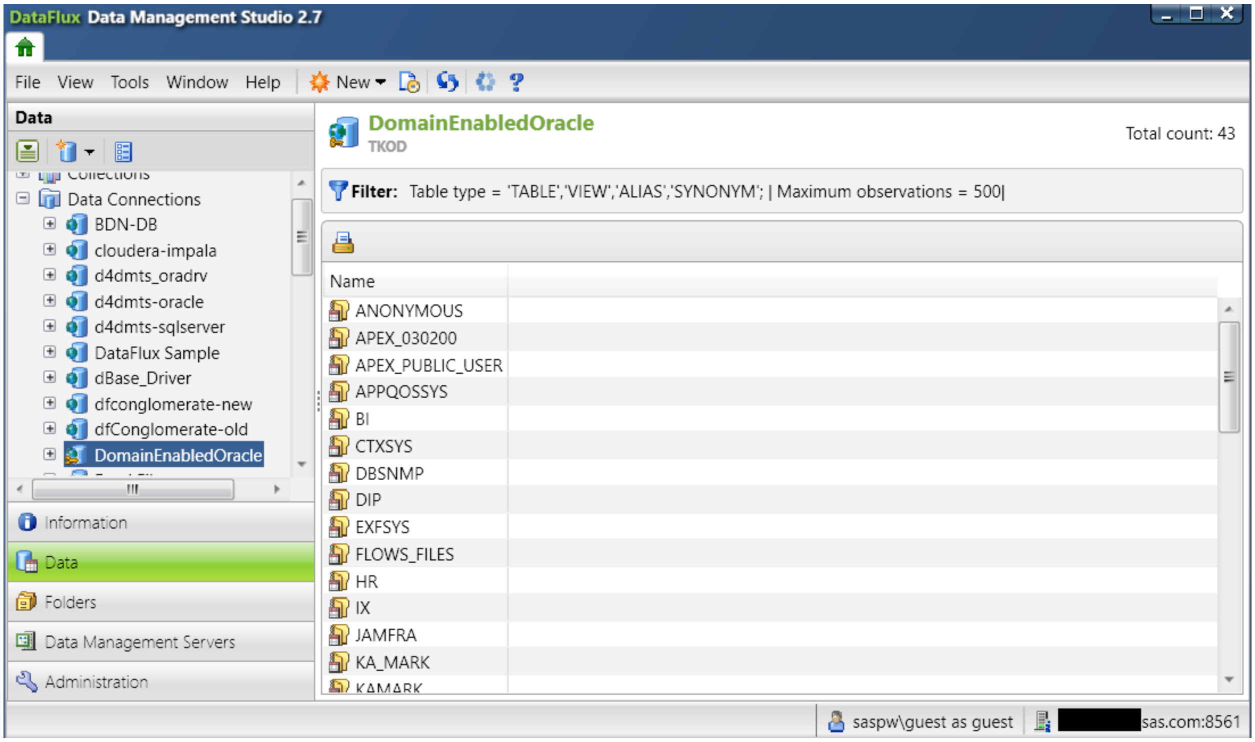
   After corrections, the attribute tag might look like the following:

   ```
   <attribute name="FILEDSN">domain_enabled_oracle.dsn</attribute>
   ```

5. If the .dftk file is not already in the network location by the DAC/DSN parameter, move the edited .dftk file to the network location.

After the .dftk file is visible in the network location, all users with DataFlux Data Management Studio configured to look for file-based DSNs on the network location via the DAC/DSN parameter see a data connection in DataFlux Data Management Studio with the name that was used during configuration of the domain-enabled ODBC connection. Users who click the data connection or use that connection in a DataFlux job automatically pick up the database user credentials from the SAS metadata. Users who do not have access to the authentication domain are prompted to provide credentials.

Display 4 shows an example in which the saspw\guest user was able to click the DomainEnabledOracle data connection and successfully connect to the database.



DomainEnabledOracle
TKOD

Total count: 43

Filter: Table type = 'TABLE','VIEW','ALIAS','SYNONYM'; | Maximum observations = 500|

Name
ANONYMOUS
APEX_030200
APEX_PUBLIC_USER
APPQOSSYS
BI
CTXSYS
DBSNMP
DIP
EXFSYS
FLOWS_FILES
HR
IX
JAMFRA
KA_MARK
KAMARK

saspw\guest as guest | [redacted]sas.com:8561

Display 4. Example of a User Connecting to the Database

The saspw\guest user did not perform any configuration steps and was not prompted for user credentials. The underlying database credentials are unknown to the saspw\guest user.

SOLUTION SUMMARY

Using a file-based DSN with a modified domain-enabled data connection greatly reduces the administrative overhead of configuring data connections and database credentials in a shared multi-user environment. The administrator can perform a one-time configuration that affects all users. Updates and changes can be performed by an administrator and those changes affect all users. Normal users do not need to configure their own data connections, and do not need to be concerned with the underlying credentials that are used to access databases. The steps to configure file-based DSNs, authentication domains, and DataFlux Data Management Studio are summarized in Appendix A.

CONCLUSION

This paper presents a methodology to implement DataFlux Data Management Studio within security boundaries that are common throughout government IT departments and elsewhere. While large-scale testing in enterprise-wide production systems is ongoing, the move toward accessing a shared DSN within a user-restricted environment worked well in the USCB-DSD pilot environment. USCB-DSD DBA staff can add databases and sources without having to contact USCB Central IT support through a ticketing system. This direct DBA control improves analytical speed and efficiency while reducing the burden on stressed IT departments.

Some questions remain after this pilot project. The biggest question is how can this solution be implemented in a cloud environment? USCB is making progress toward a full-cloud IT infrastructure and implementing DMS will be a challenge. However, because cloud providers such as Amazon Web Services continue to innovate solution architectures and SAS continues to add data-management functionality to SAS® Viya®, robust migration pathways from on-premises solutions are expected.

REFERENCES

United State Census Bureau. 2019. “Census at a Glance.” Accessed February 10, 2019. **https://www.census.gov/about/what/census-at-a-glance.html**

United State Census Bureau. 2019. “Privacy and Confidentiality.” Accessed February 20, 2019. **https://www.census.gov/history/www/reference/privacy_confidentiality/privacy_and_confidentiality_2.html**

ACKNOWLEDGMENTS

The authors would like to thank Karuna Plachco, Lisa Rich, and Rayna Rowell for reviewing this paper.

RECOMMENDED READING

- DataFlux® Data Management Studio 2.7: User Guide
- DataFlux® Data Management Studio 2.7: Installation and Configuration Guide

CONTACT INFORMATION

Marc Price
SAS Institute, Inc
support@sas.com

APPENDIX A

Here is a summary of the one-time setup required by administrators to implement the solution described in this paper:

1. Edit the <*install dir*>\etc\app.cfg and add the following:

   ```
   DAC/DSN=\\writable_network_drive\path
   ```

2. Edit the <*install dir*>\etc\ui.cfg and add the following:

   ```
   BASE/OMR_PROMPT_ON_STARTUP=TRUE
   BASE/AUTH_SERVER_LOC=iom://metadataserver.yourcompany.com:8561
   ```

3. Run the Windows 32-bit ODBC Data Source Administrator, click **File DSN** tab, and create a DSN as usual. Windows prompts you for a file name.
4. Copy the .dsn file created in step 2 (which is likely in My Documents or Documents folder) to the same location specified in step 1 on the DAC/DSN parameter.
5. Create an authentication domain in SAS Management Console and add credentials in that domain on the **Accounts** tab for any users or groups. If you need to save the credentials on a group, add users as members of that group.
6. Create a domain-enabled ODBC connection in DataFlux Data Management Studio using a regular ODBC DSN as described in the "Adding Domain Enabled ODBC Connections" section in *DataFlux Data Management Studio 2.7: User Guide*.
7. Edit the generated .dftk file for the domain-enabled ODBC connection created in the previous step. The .dftk file is in the location specified in step 1 on the DAC/DSN parameter.
8. Verify that the correct authentication domain appears in the domain tags.
9. In the name attribute, change name="`DSN`" to name="`FILEDSN`".
10. In the name attribute, specify the path and file name for the DSN file that was copied to a shared location.