

Use internal SAS® metadata user and authentication domain to connect to an FTP server.

Sebastian Scanzi, S4S Consulting Ltd, UK

Abstract.

A SAS® based application was implemented in a SoX¹ environment by a bank. The system requires up to date data as input from different sources (internal and external). One of the external data source was held on an external third party FTP server and the last available working day data of each month needed to be downloaded.

One of the requirements for the FTP process was that the transfer of data needed to be automated and to be SoX compliant. The latter point was the most challenging one as the main FTP credentials could not be coded within the process itself. The normal approach of having the FTP password encrypted with PROC PWENCODE in conjunction with filename FTP engine was not suitable.

A series of options were considered (ie: Oracle wallet) but because of time and constraints none appeared viable for this project. A different approach needed to be taken and because the SAS Metadata was available to us we decided to use the SAS Authentication Domain approach.

One important point was that none of this would be required if we could simply store the passwords in open code. But that would not be secure. So the challenge was to find a way to automate the connection to the external FTP site using specific credentials, but to have these credentials visible or accessible only to the process we were going to use. But they had to be stored somewhere because we were not going to type them in manually every time we connected.

Introduction.

Base SAS provides a lot of ways to connect to an FTP server. A lot of documentation and papers are available on how to connect to an FTP server using Base SAS. This paper is going to concentrate on how to connect to an FTP server using a SAS Authentication Domain.

In our scenario, we have a few problems that need to be sorted out:

1. The service account which is used to run the application, cannot be setup as a SAS user within the SAS Metadata because the password changes every time the glass break through process is invoked to access the service account's password.
2. The FTP credentials cannot be stored in any way within the application (ie: SAS code, scripts)

¹ SoX: The Sarbane-Oxley Act.

The problem and the approach taken.

To connect to an FTP server the following syntax is usually used:

```
filename _ftp ftp "&file" cd="/" user="&USERNAME" pass="&PASSWORD"  
host="&SERVER";
```

The above statement defines a fileref “_ftp”. “&file” is a macro variable which resolves to the file to be downloaded. “&username” and “&password” are login credentials to the remote FTP server.

The idea is to remove the user= and pass= from the filename syntax. To achieve this is to create a new authentication domain within the SAS metadata and then link the FTP credentials with the newly created authentication domain.

The syntax will look something like:

```
filename _ftp ftp "&file" cd="/" host="&SERVER" authdomain="FTPAuth";
```

To access the SAS metadata authentication domain a user needs to be defined within the SAS metadata. In our scenario using the service account credentials is pointless. The password is changed every time we access the service account’s password through break glass.

Connecting to the SAS metadata the following syntax is used:

```
options metaserver="win2008r2"  
        metaport=8561  
        metaprotocol="bridge"  
        metauser="USERID"  
        metapass="PASSWORD";
```

Because of the security constraints we need to make sure the access to the application is not compromised. Creating a new domain/local account to allow the service account to access the SAS metadata will not resolve the problem. A domain/local account is usually linked to a physical person and will have some kind of access to the network. First the IT department will not create a domain/local account to be linked to a service account. Second the password will have to be stored somehow within the application. If the domain/local account gets compromised, then this opens a backdoor to potential damages to the network and the application.

As we solely want to access the SAS metadata to retrieve the FTP credentials stored within the SAS authentication domain, the creation of a SAS internal account within the SAS metadata is the way forward. The creation of such an account presents a multitude of benefits:

- The internal user is defined directly within the SAS metadata. The SAS admins create the user. The IT department is not involved.
- The internal user has only access to SAS metadata not to the domain nor the operating system.
- The internal user cannot login to any systems.
- The internal user does not follow the security policy set up by the IT department (ie: password change every 60 days).

When using a SAS metadata internal account, @saspw suffix needs to be used. For the purpose of this paper the internal user we use is test_admin, the syntax will be **test_admin@saspw**.

Below is the syntax on how to access a file on a FTP server using test_admin internal user to retrieve the FTP credentials (we use the user waka to connect to the FTP server).

```
options metaserver=localhost
      metaport=8561
      metauser="test_admin@saspw"
      metapass="&password"
      metarepository=Foundation;

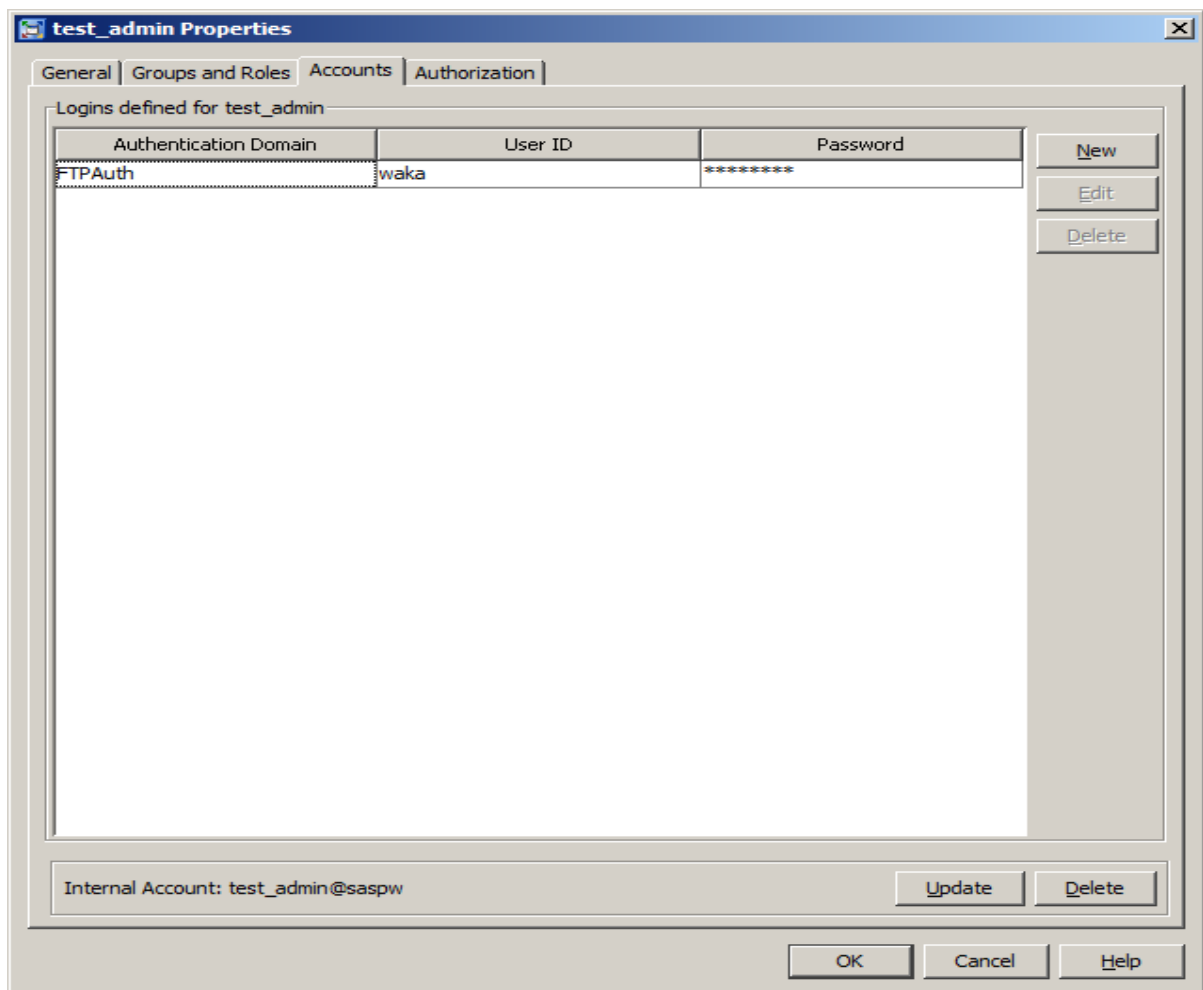
filename test ftp '_test.txt' cd='/Data/test_ftp' host="&host"
authdomain="FTPAuth" debug passive;

data test;
infile test;
length var1 8.;
input var1;
run;
```

SAS log:

```
NOTE: 220 DiskStation FTP server ready.
NOTE: <<< 220 DiskStation FTP server ready.
NOTE: >>> USER waka
NOTE: <<< 331 Password required for waka.
NOTE: >>> PASS XXXXXX
NOTE: <<< 230 User waka logged in.
NOTE: >>> PASV
NOTE: <<< 227 Entering Passive Mode (XXX,XXX,XXX,XXX,XXX,XXX)
NOTE: >>> TYPE A
NOTE: <<< 200 Type set to A.
NOTE: >>> CWD /Data/test_ftp
NOTE: <<< 250 CWD command successful.
NOTE: >>> PWD
NOTE: <<< 257 "/Data/test_ftp" is current directory.
NOTE: >>> RETR _test.txt
NOTE: <<< 150 Opening ASCII mode data connection for '_test.txt' (13 bytes).
NOTE: User waka has connected to FTP server on Host .
```

In SAS Metadata:



Conclusion:

The use of a SAS Metadata internal user combined with the authentication domain provides a lot of benefits:

- Password and security can be managed by SAS Admins
- Low maintenance and straight forward setup process.
- Password is protected and not visible to users
- Password can have regular updates without interference to users.

In summary, there are ways of doing this using other technologies (i.e:Oracle wallet) but this demonstrates that SAS has its own capability of managing external credentials securely.

DISCLAIMER:

The contents of this paper are the work of the author and do not represent the opinions, recommendations, or practices of SAS Institute.

ACKNOWLEDGMENTS

The author would like to thank Andrew Wilcox, Phil Baker and Toby Hill for their contributions to this paper

CONTACT INFORMATION

Please contact the author if you have any questions or comments:

Mr Sebastian Scanzi - S4S Consulting Limited

Email: sebastian.scanzi@s4sconsulting.co.uk

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.