

Paper CT-26
Securing Your SAS Systems
 - A Simple Step to Identify Users -

Leanne Tang, National Agricultural Statistics Service USDA, DC

ABSTRACT

Securing our applications and data has become an integral part of our SAS® systems. Identifying users and what they are entitled to is the first step in this effort. Our organization utilizes Microsoft Active Directory Services for user authentication and authorization. DSQuery is a command-line utility from Microsoft which is part of Windows Server 2003 and later operating systems. DSQuery can be executed from a SAS program to query Active Directory for user account information including group memberships. This information can then be used from within SAS programs to authorize users. This paper will discuss some common features of the DSQuery utility and how to use query result in SAS programs to validate users.

INTRODUCTION

When SAS programs have to access multiple remote databases and be executed by users from different organizational divisions and locations, data security becomes a major concern. In one of my SAS applications, I need to identify the organizational unit for each user. Based on the organizational unit, a subset of data is queried for review and analysis in the SAS programs.

One way to achieve this is to create and maintain a table of users and their organizational unit. This seemingly simple task, however, can become very daunting in an organization with hundreds of employees in multiple office locations where employees are regularly changing positions. It creates a challenge to maintain a static list from a moving target. After talking with the information technology (IT) staff, I learned that the information required was already available in Active Directory, and this information is maintained and updated by IT personnel. We can rely on every user of the SAS application having an Active Directory domain user account since it is required for logging into the organization's network in order to execute SAS programs.

I obtained a copy of DSQuery from our IT server team and placed it in a directory accessible by SAS users. The rest of this paper will focus on how to use the DSQuery utility to retrieve the users' organizational unit information for my SAS programs.

ACTIVE DIRECTORY AND DSQUERY

Microsoft Active Directory is used for managing users, computers, groups, printers, applications, and other directory-enabled objects. Furthermore, groups defined in Active Directory can be used to grant and control access to local and remote file system resources such as shared network drives. Active Directory runs as a series of services on specialized servers called domain controllers. When a workstation, server, user, executable, or SAS program queries Active Directory, it is a domain controller that responds to the request.

DSQuery is a command-line utility installed with Windows Server 2003 and newer operating systems or as part of a Windows Server Resource Kit which can be downloaded from the Microsoft website. The utility is generally used by server administrators for searching Active Directory for specific object types such as computer, contact, subnet, group, organizational unit, site, server, user, etc.

By default, every user with an Active Directory account can access and query AD objects using the DSQuery utility. The basic command line syntax for query user information is:

```
dsquery user [{<StartNode> | forestroot | domainroot}] [-o {dn | rdn | upn |
samid}] [-scope {subtree | onelevel | base}] [-name <Name>] [-desc
<Description>] [-upn <UPN>] [-samid <SAMName>] [-inactive <NumberOfWeeks>] [-
stalepwd <NumberOfDays>] [-disabled] [{-s <Server> | -d <Domain>}] [-u
<UserName>] [-p {<Password> | *}] [-q] [-r] [-gc] [-limit <NumberOfObjects>]
[{-uc | -uco | -uci}]
```

DSQuery can also be used to query other objects in Active Directory such as computer, partition, or server, but this paper will only discuss querying user objects. For information on querying other objects see the Microsoft online library <http://msdn.microsoft.com/en-us/library>.

USING DSQUERY UTILITY IN SAS PROGRAMS

The following program is used to get a user's organizational unit information from Active Directory.

First, I assigned the location of the DSQuery utility to a SAS macro variable named myDSQuery. In the FILENAME statement, the device-type key word PIPE is used. This is to invoke a program that is external to SAS and redirect the program's output back to the SAS program without creating any intermediate data file. The host-option CONSOLE=MIN indicates that the command prompt window that is opened to run DSQuery program is opened minimized.

The parameter, -samid, specifies to search for any account with the SAM ID equal the SASUSERID. SASUSERID is an automatic global macro variable created by the macro processor when the SAS is invoked.

```
%LET myDSQuery = H:\Hqapps\SASEIS\CommonDev\Programs\dsquery.exe ;
FILENAME myDN PIPE "&myDSQuery user -samid &SYSUSERID" console=min;
DATA _NULL_;
    INFILE myDN MISOVER ;
    INPUT myDN & $255.;
    CALL SYMPUT('myDN', myDN);
    PUT myDN= ;
RUN ;
```

After the DSQuery utility is executed using a FILENAME statement with the key word PIPE, the DATA step is executed to read the DSQuery output into SAS and assign the value to a macro variable. The SAS log below shows the macro variable, myDN, value in a long string.

Output 1. The DSQuery Result:

```
1 %LET myDSQuery = H:\Hqapps\SASEIS\CommonDev\Programs\dsquery.exe ;
2 FILENAME myDN PIPE "&myDSQuery user -samid &SYSUSERID" console=min;
3 DATA _NULL_;
4     INFILE myDN MISOVER ;
5     INPUT myDN & $255.;
6     CALL SYMPUT('myDN', myDN);
7     PUT myDN= ;
8 RUN ;
```

NOTE: The infile MYDN is:

Unnamed Pipe Access Device,
PROCESS=H:\Hqapps\SASEIS\CommonDev\Programs\dsquery.exe user -samid TangLe,
RECFM=V,LRECL=256

myDN="CN=TangLe,OU=ees,OU=smb,OU=SD,OU=HQ,DC=nassad,DC=nass,DC=usda,DC=gov"

NOTE: 1 record was read from the infile MYDN.

The minimum record length was 70.

The maximum record length was 70.

NOTE: DATA statement used (Total process time):

real time	0.71 seconds
cpu time	0.07 seconds

PARSING THE INFORMATION

The second step is to reformat the information from a single string to the separate units of values that are usable in the SAS program. A DATA step with DO loop is used to scan through the string. The individual values are stored in the userInfo table.

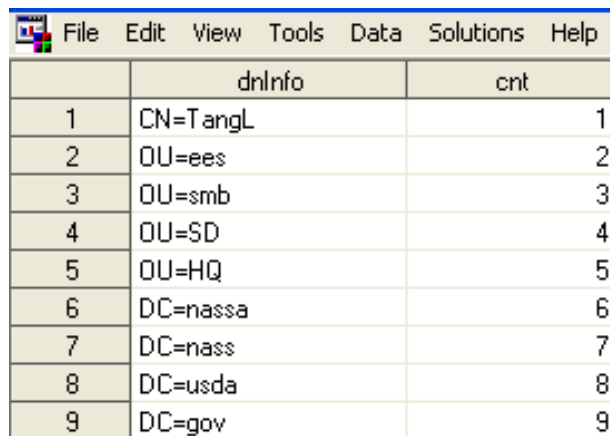
```

DATA userInfo (KEEP = CNT dnInfo) ;
  LENGTH dnInfo $20. ;
  itemMax = COUNTW(&myDN) ;
  DO CNT = 1 TO itemMax ;
    dnInfo = SCAN(&myDN, CNT, ',', 'MO') ;
    OUTPUT ;
  END ;
RUN ;

```

The output table is displayed below. The first observation is the user ID. The second to ninth observations are the organizational units associated with the user ID.

Table 1. Parsed user information



	dnInfo	cnt
1	CN=TangL	1
2	OU=ees	2
3	OU=smb	3
4	OU=SD	4
5	OU=HQ	5
6	DC=nassa	6
7	DC=nass	7
8	DC=usda	8
9	DC=gov	9

CONCLUSION

There are certainly other ways to complete this task, but I have found this method to be very efficient and effective. DSQuery bridges the information between Microsoft Active Directory and SAS. This opens up new possibilities and provides SAS programmers more opportunities to work efficiently.

While this paper only discussed using DSQuery utility to search for user objects, there are many other objects which can be queried with DSQuery. Furthermore, by using the FILENAME PIPE method, there are many other Windows executables which can be explored in the future.

The SAS programs in this paper are tested using SAS 9.2.

REFERENCES

Microsoft Windows Server Library. Available at [http://technet.microsoft.com/en-us/library/cc732952\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc732952(v=ws.10)).

Using Unnamed Pipes from SAS(R) 9.2 Companion for Windows, Second Edition. Available at <http://support.sas.com/documentation/cdl/en/hostwin/63285/HTML/default/viewer.htm#unnamed.htm>

Varney, Brian. 2008. Check out These Pipes: Using Microsoft Windows Commands from SAS®. Available at <http://www2.sas.com/proceedings/forum2008/092-2008.pdf>.

ACKNOWLEDGMENTS

Many thanks to Troy Hamilton for providing the knowledge and support on using DSQuery utility.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Leanne Tang
National Agricultural Statistics Services, USDA
1400 Independence Ave., SW
Washington, D.C 20250-2001

202 – 720 6957

Leanne_tang@nass.usda.gov

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 trademarks of their respective companies.