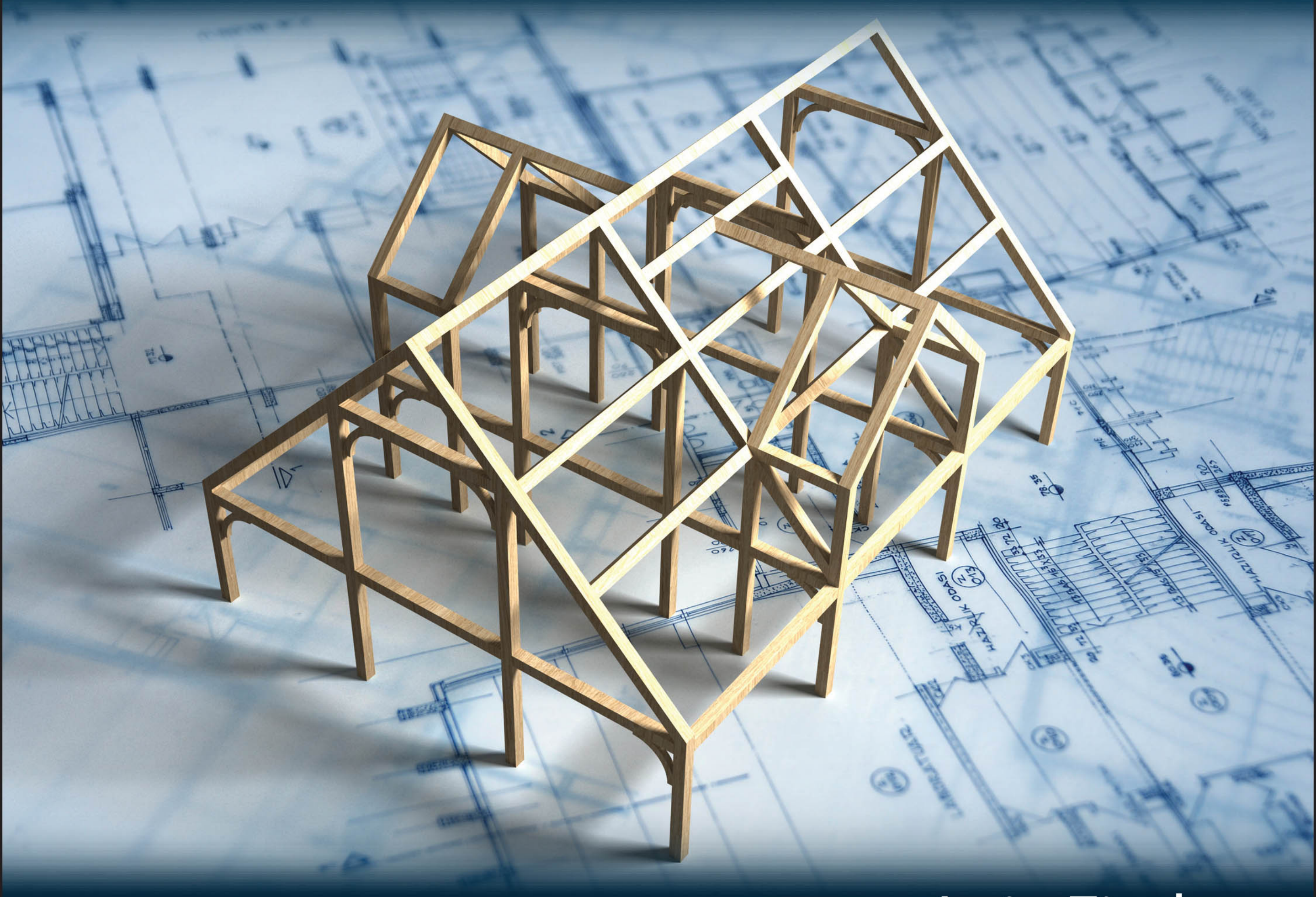
SAS® Administration from the Ground Up

Running the SAS®9 Platform in a Metadata Server Environment



Anja Fischer

The correct bibliographic citation for this manual is as follows: *Fischer, Anja*. 2019. *SAS® Administration from the Ground Up: Running the SAS®9 Platform in a Metadata Server Environment*. Cary, NC: SAS Institute Inc.

SAS® Administration from the Ground Up: Running the SAS®9 Platform in a Metadata Server Environment

978-1-64295-191-2 (Hardcover)
978-1-63526-313-8 (Paperback)
978-1-63526-729-7 (Web PDF)
978-1-63526-727-3 (epub)
978-1-63526-728-0 (mobi)

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

March 2019

Contents

About This Book

What Does This Book Cover?

This book is about the basic principles of SAS 9.4 platform administration. It is a starter guide for new SAS administrators, helping you to turn into a happy, calm and confident SAS admin. The book provides you with a light entrance into the world of SAS administration, without wading through the documentation. This book does not cover SAS Viya administration.

Is This Book for You?

If you are new to the SAS 9.4 admin job or are an advanced admin who wants to make sure you know all the admin basics and admin tricks, this book is for you.

What Should You Know about the Best Practices?

This book provides recommendations and best practices around the most important SAS 9.4 platform administration topics a SAS Platform administrator should be familiar with.

Additional Resources

Please find an appendix full of handy links and references on the author's page at http://support.sas.com/fischer.

We Want to Hear from You

Do you have questions about a SAS Press book that you are reading? Contact us at saspress@sas.com.

SAS Press books are written *by* SAS Users *for* SAS Users. Please visit sas.com/books to sign up to request information on how to become a SAS Press author.

We welcome your participation in the development of new books and your feedback on SAS Press books that you are using. Please visit sas.com/books to sign up to review a book

Learn about new books and exclusive discounts. Sign up for our new books mailing list today at https://support.sas.com/en/books/subscribe-books.html.

Learn more about this author by visiting her author page at http://support.sas.com/fischer. There you can download free book excerpts, access the appendix, read the latest reviews, get updates, and more.

Chapter 4: Do I Know You? A Quick Review on Users, Groups, and Roles in SAS 9.4

Introduction to User, Group and Role Management

In this chapter, I would like to talk about users, groups and roles in SAS. The SAS documentation gives a good introduction on how to create users, groups and roles, so let's talk a bit about what users, groups and roles in SAS are and how they work. Understanding the user, group and role management is important when it comes to building your SAS environment.

Let's start by leaving SAS for a moment and just think about users and groups from an authentication (domain) perspective in general. Users and groups can be created either

- On a local machine, or/and
- On a computer network using directory service accounts such as LDAP or Active Directory

You create users and groups to "open the doors" for your users: they can log on to your system, run code, access data that you store in system folders, etc.

Now let's turn to SAS and talk specifically about users and groups in SAS. In SAS, you create users and groups in metadata, using SAS Management Console or SAS Environment Manager. By creating users and groups in metadata, you, too, open the door for users, just this time the door for the SAS metadata world.

> **Note:** You can use SAS clients, such as SAS Enterprise Guide, SAS Studio, etc., *without* creating users in SAS. Your users just simply use their OS credentials.

However, to use all the features in SAS 9.4, especially security, and to use reporting tools such as SAS Environment Manager, you must have users in SAS metadata. As a best practice, SAS recommends creating and maintaining your users and groups in metadata.

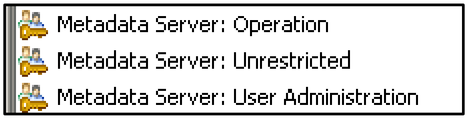
Aside from users and groups, as mentioned in the title of this chapter, there are also roles. Before we will dive deeper into talking more about users and groups, let's check off the roles.

Roles

As an example of a role in a different context: User Gwendolyn has an access card to enter a building, but her access card doesn't give her the capability to go on the floor where the finance department is located. Benji is a manager, so his access card provides the capability to access every floor and every department in that building. If we take that example and apply it to SAS, Gwendolyn might have the capability to use certain features in a SAS client, whereas user Benji might have the capability of using all features of a SAS client. Even SAS administrators can have different capabilities, which comes in handy for a team of admins, where each admin fulfills a different ***role***.

To give you an example, let's look at some of the admin roles in SAS and the different capabilities that they provide:

Figure 4.1: Admin Roles



Metadata Server: Operation
Metadata Server: Unrestricted
Metadata Server: User Administration

Metadata Server: Operation. Admins can create custom and project repositories and operate the metadata server (start, stop, pause, resume, quiesce), using the Server Manager plug-in in SAS Management Console.

Metadata Server: Unrestricted. Admins with this role applied have all capabilities and full access to metadata. However, they cannot read users' passwords.

Metadata Server: User Administration.
Admins can create and manage non-admin users, groups and roles, internal accounts, logins and authentication domains

Important key points about metadata-based application roles

- Roles do not protect data or metadata. This is an important point and often misunderstood. Roles just control which features in a particular application are available to which users.
- An application feature that is under role-based management is called a capability. Each role provides multiple capabilities. A user or group can be in multiple roles, such as access to Server Manager, access to certain functions in SAS Enterprise Guide, etc.
- Not all applications have roles. Not all application features are under role management. Each application that supports roles provides a fixed set of

capabilities. You cannot convert a feature that is not a capability into a capability. If you have the need to apply capabilities that are not part of a role, it is a best practice (a very strong recommended best practice), to create new custom roles and apply the capabilities you need to this new role. If you use an already existing role, it defeats the purpose, as every already existing role fulfills a certain purpose. If you modify it, you change the roles' role that it plays in the SAS environment. Remember it that way: every role has a role to play, each of which is different. In a cast of a play you won't find one and the same characters twice on stage, if that makes sense.

> **Note**: Custom Tasks in SAS Enterprise Guide:
> You have the option to register custom tasks as capabilities in SAS Enterprise Guide. Generally, you can allow or disallow custom tasks. If you allow them, you can restrict these custom tasks to just approved tasks. The following blog is a great resource:

Controlling access to custom tasks in SAS Enterprise Guide at https://blogs.sas.com/content/sasdummy/2013/01/07/controlling-access-to-custom-tasks-in-sas-enterprise-guide/

- Capabilities are additive. There are no negative capabilities (capabilities that limit what a user can do). It is not possible to deny a capability (capabilities are either granted or not granted).
- Capabilities can be categorized as follows:
  - explicit capabilities
    Can be incrementally added to or removed from any role (other than the unrestricted role, which always provides all explicit capabilities). Most roles have explicit capabilities.
  - implicit capabilities
    are permanently bound to a certain role. The metadata server's roles provide implicit capabilities. For example, the user administration role provides the capability to add users, but there is no explicit *Create Users* capability.
  - contributed capabilities
    are implicit or explicit capabilities that are assigned through role aggregation. If you designate one role as a contributing role for another role, all of the first role's capabilities become contributed capabilities for the second role.
- Important: You cannot assign permissions to a role. You cannot assign capabilities to a group.

  A user cannot temporarily assume or relinquish a role. All of a user's roles are active at all times.

For more resources about roles can be found in the Appendix.

OK, now we can check off the roles. Next, lets dive into the world of users and groups in SAS.

Users

Let’s get back to users and groups. The SAS Management Console: Guide to Users and Permissions explains it in the following way:

In order to make access distinctions and track user activity, security systems must know who is making each request.

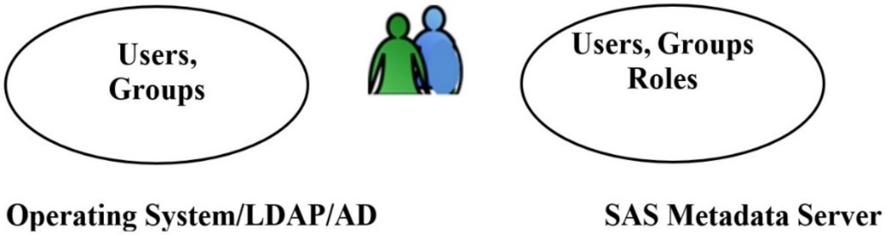
This goes back to what I mentioned earlier, that you have to have users in metadata in order to apply metadata security, in order to do reporting, auditing and monitoring using SAS Environment Manager, etc.

So, the main purpose of user administration is to provide information that helps SAS make this determination. The central piece of user information that the SAS environment requires is **one external account ID for each user**. The SAS environment uses its copy of these IDs to establish a unique SAS identity for each connecting user. All of a user's group memberships, role memberships, and permission assignments are ultimately tied to their SAS identity.

If you do not want to create your users in metadata interactively by using the User Manager plugin in SAS Management Console, you can write a program that performs these tasks as batch processes. You can do this using the user import macros. A link to the documentation is in the Appendix.

> **Note:**
> For identification purposes, only the account IDs are needed. SAS does not maintain copies of external passwords for identification purposes.
>
> Leaving the official doc, let me continue in the Anja-style. For the OS users and SAS users, see Figure 4.2.

Figure 4.2: A visual depiction of OS users and SAS users

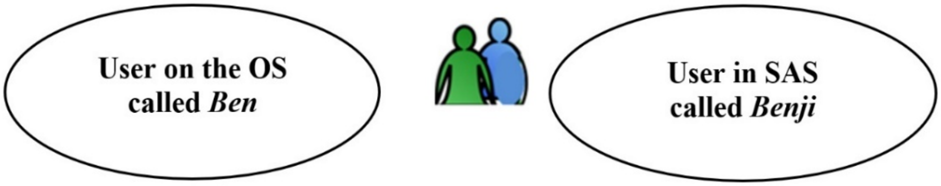


OS/LDAP/AD/Database accounts are on the left in the oval figure. And SAS accounts are on the right.

Example

If we create a user in SAS metadata using SAS Management Console’s or SAS Environment Manager’s User Manager plug-in, that is associated with an external user ID, it could look like Figure 4.3.

Figure 4.3: A visual depiction of OS user Ben and SAS user Benji



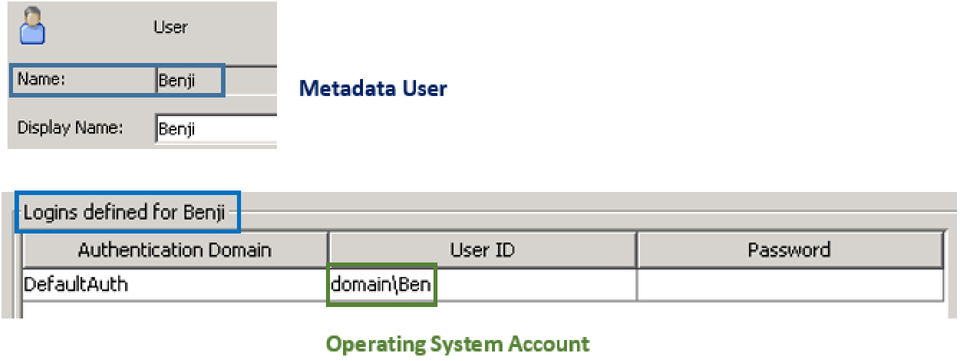
SAS user Benji uses OS user Ben in the background, meaning, user Benji is associated with user Ben.

In a more professional manner, this is the official description of a SAS user definition (aka login):

A login is a SAS copy of information about an external account. Every login must include a user ID. In a login for a Windows account, the ID must be qualified (for example, user@company.com), domain\user, or machine\user.

So, in our case, using **Ben** as the OS account and **Benji** as the SAS account, the SAS user ID in SAS Management Console or SAS Environment Manager, *User Manager plug-in*, would look like Figure 4.4.

Figure 4.4: Metadata and OS user



User

Name: Benji

Display Name: Benji

Metadata User

Logins defined for Benji

Authentication Domain	User ID	Password
DefaultAuth	domain\Ben	

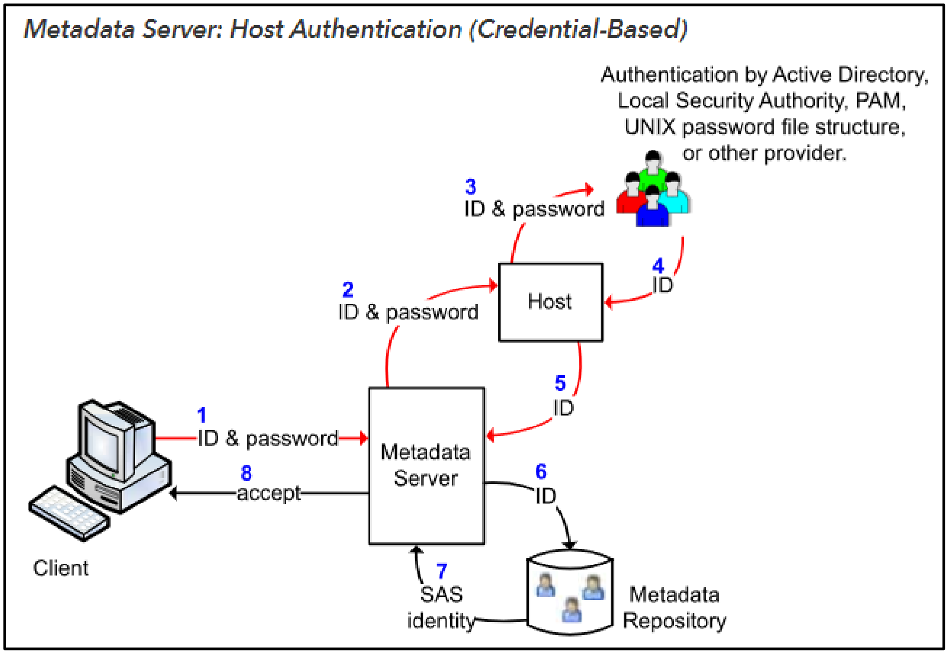
Operating System Account

The explanation above is important when it comes to authentication. When a user logs on to a SAS client, such as SAS Enterprise Guide, the SAS user ID is verified in the background to make sure the users are who they say they are.

Note: You must log on as an admin user to be able to work with user IDs. You can either use the unrestricted user ID sasadm@saspw or an admin user that has the capability to manage users. This capability is given with a role. An example for a role that provides this capability is *Metadata Server: User Administration*

So, going back to users, SAS identities, OS accounts and authentication, Figure 4.5 shows nicely how it works.

Figure 4.5: Host Authentication.


Metadata Server: Host Authentication (Credential-Based)
Authentication by Active Directory, Local Security Authority, PAM, UNIX password file structure, or other provider.
3
ID & password
4
ID
2
ID & password
Host
5
ID
1
ID & password
8
accept
Metadata Server
6
ID
Client
7
SAS identity
Metadata Repository

When a user launches a SAS client, an authentication process with two phases occurs:

In the verification phase, the system ensures that the user is who he or she claims to be. An example for a credential-based host authentication is as follows:

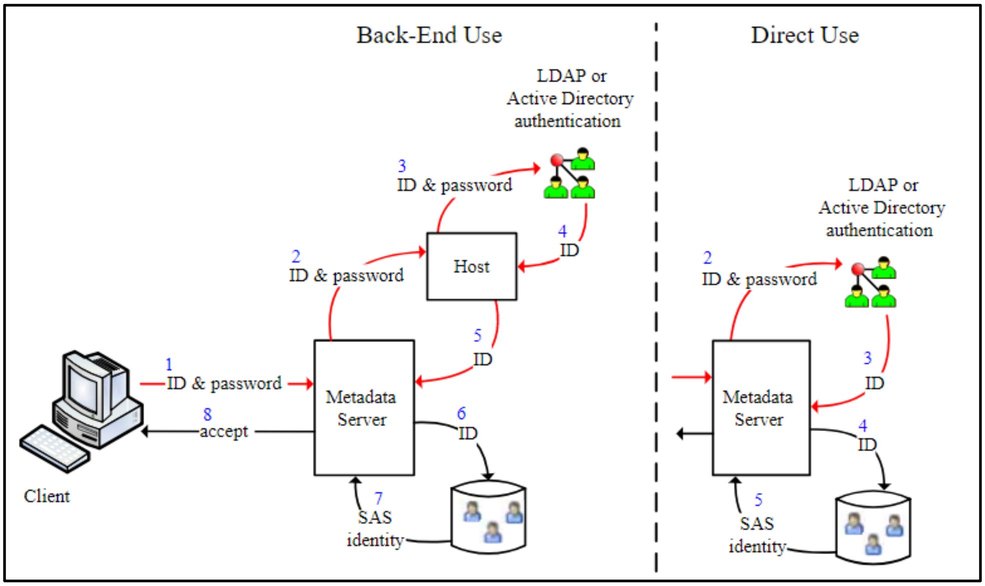
- The client prompts the user for an ID and password.
- The user enters credentials that are known to the metadata server's **host**.
- The client sends the credentials to the metadata server.
- The metadata server passes the credentials to its host for authentication.
- If the host determines that the user has a valid account, the host returns the authenticated user ID to the metadata server.

In the SAS identity phase, the system resolves the authenticated user ID to a particular SAS identity. In this phase, SAS examines its copies of user IDs in metadata in an attempt to find one that matches the authenticated user ID. Simply put: Once the user is authenticated by the operating system, the metadata server goes through the users in the User Manager and checks whether there is a metadata identity that has the successfully authenticated user login with the matching metadata user ID.

If a matching user ID is found, a connection is established under the owning identity. The owning identity is the user or group whose definition includes an operating system login with the matching metadata user ID.

If we throw an additional authentication provider in the mix, such as LDAP, the following Figure 4.6 shows a comparison of authentication.

Figure 4.6: Metadata Server



More information about the link to the different authentication mechanisms can be found in the Appendix.

What if the metadata server cannot find a matching user in metadata (User Manager)?

When SAS is installed, two implicit groups are created by default for you: PUBLIC and SASUSERS. If the metadata server cannot find a metadata user in the User Manager that stores the successfully authenticated OS identity, the user is authenticated as a so called **public-only** user. Literally. A public user does not belong to a particular group and does not have a particular user ID and is therefore considered a general person without any metadata user affiliation.

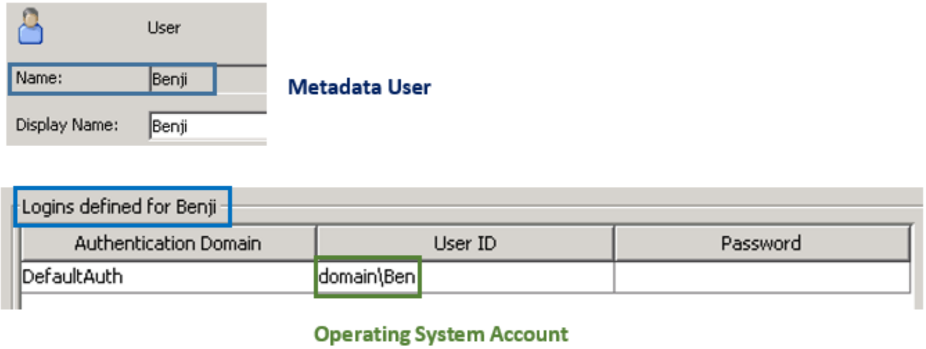
So, lets dig deeper into our user and group adventure and let’s talk about PUBLIC and SASUSERS.

As mentioned prior, when SAS is installed, two so called **implicit** groups are created by default for you: PUBLIC and SASUSERS.

PUBLIC and SASUSERS exist in every metadata server environment, no matter what products you have licensed.

SASUSERS includes all successfully authenticated OS users that are tied to a metadata user ID. A user successfully outside-of-SAS-authenticated with a metadata identity (a user that you see in the User Manager) makes a user member of the SASUSERS group.

So, if you have the scenario:



User

Name: Benji

Display Name: Benji

Logins defined for Benji

Authentication Domain	User ID	Password
DefaultAuth	domain\Ben	

User Benji is initially identified as a member of SASUSERS.

If you don't have the above, if you don't see a user ID for that user in metadata, the user is authenticated as PUBLIC. Just keep this rule in mind whenever you work with users and groups:

A user successfully-outside-of-SAS-authenticated with a metadata identity (a user that you see in the User Manager) makes a user member of the SASUSERS group.

A user successfully-outside-of-SAS-authenticated without a metadata identity (user is not in the User Manager) makes a user a member of the PUBLIC group.

Now, knowing this, think about it in terms of security: Would you want to let just everyone into your house or only people that you know? It's the same concept. I don't want users that are not known to the metadata server messing with my metadata. I want to control who accesses my assets, and for that reason, I make sure that PUBLIC users will never be able to get into my environment. Just keep this in mind for now. You will need this when we talk about security later on.

Note: There are always exceptions to the rule, and it might be useful in your case to allow public access. The general rule is to avoid it. This would have to be decided case by case and should be very well thought through.

At this point, you might be asking yourself: If users that are created in metadata and are associated with a successfully authenticated OS user ID, why are they put into a “non-explicit-metadata-user” group called SASUSERS? And, if SASUSERS means there is a metadata identity, wouldn’t that make it explicit?

These are great questions. To address these questions, we move on to groups in SAS.

Groups in SAS (SAS User Groups)

To recap, we identified users in the User Manager that are associated with an OS account as members of an implicit group called SASUSERS. Let’s assume you have 500 individual users. Would you want to assign permissions on objects 500 times – for each individual user? Or, would you rather minimize the effort and use groups to assign permissions to objects? I would definitely choose the latter!

Let me give you a good group example:

I am working in a department that is called *Customer Success*. The department consists of two different teams:

We have a team of :

1. *Systems Engineers* – handling all things technical (I am a member of this team)
2. *Customer Success Managers* – who focus on post-sales activities.

Rather than giving permissions to each individual Customer Success team employee, the employees have been put into **groups**. All groups in our department are: A group for System Engineers, a group for Customer Success Managers, a group for managers, and groups for each business unit the Customer Success Managers work in.

So, you can see, all employees have been placed into groups based on:

- job description and tasks, the type of work
- data we each need to access, and
- responsibilities, etc.

Data and information our managers access might not be accessible to us employees; technical info for Systems Engineers might not be that important to Customer Success Managers, and so forth. To make things easier and distinct, these *explicit* groups have been created.
Yes, of course, you must create individual users first to be able to create groups, but the end goal for creating users in metadata is to minimize the administration and maintenance effort as much as possible by using groups.

Going back to the questions about why SASUSERS is being called an implicit group even though it seems to be explicit. Every user in your SAS environment is member of SASUSERS, and remains a member of SASUSERS, even if put into an explicit group. Going back to my previous Customer Success Department example, each individual member of the Customer Success department is a member of SASUSERS.

When I first started as an admin, many moons ago, I got dizzy with all the implicit and explicit, but it is really pretty simple if you think about it in terms of the example:

Benji is a Systems Engineer. When he logs on to SAS, he is authenticated as member of SASUSERS. But, on top of that, the metadata server is smart and realizes that there is another group for Benji, called Systems Engineers. And even so Benji remains in the SASUSERS group, the permissions that might be explicitly set for the custom group Benji is a member of, take precedence.

SASUSERS and PUBLIC will come into play when we talk about permissions later. For now, it is just important that you understand the concept of users, groups, roles, explicit and implicit groups.

Bottom line:

In summary, once you create individual users, you can then create groups and organize your users into groups. A user can belong to multiple groups; however, this must be carefully thought through. Interestingly, the question on whether to create users first or groups first, seems to kick of passionate discussions at times. To me, it fits into the "what was first, the chicken or the egg" category, and is a rather philosophical question each customer has to answer for themselves. If you are wondering whether to create users first or groups first, I think it is easier to bring all my users into my SAS environment and then create groups based on my users. Either way, the main ingredient for a successful user and group set up is to create a concept, a plan, beforehand. It does not make sense to come up with wild group names if they don't fulfill a purpose. I would not create a group called Dogs and add Cats to it.

The best advice I can give you:

Think about your user base. Are you responsible for one department, one team, or several? If the latter, does it make sense to create groups based on team names, did someone else before you maybe already create users? And further, do some users access the same data? Do users share data? What tasks do your users fulfill? What SAS clients are they using, what output are they creating, and so forth. These are examples for criteria that you should use before you put groups into place.

Whether you decide to create groups first or, users first, – what matters is that you have a good strategy in place.

Good-to-Know: What if I Already Have Users and Groups in LDAP or AD?

If you use LDAP or AD, SAS provides you with an option to import your LDAP/AD users and groups into SAS metadata (they will appear in the SAS Management Console's User Manager). Check out *Usage Note 40628: Automating the addition of users and groups to a SAS® Metadata Repository*, at: http://support.sas.com/kb/40/628.html (that's a short link to type, right?)

Let’s summarize what we have covered so far about user, groups, and roles:

- It is a best practice to create users and groups in metadata using the User Manager plug-in in SAS Management Console or SAS Environment Manager, to synchronize users and groups with an enterprise directory like Active Directory or LDAP.
- Metadata user IDs are associated with operating system/LDAP/AD or database accounts.
- Metadata users are authenticated with the operating system before getting access to metadata.
- After successful authentication on the OS, users are considered members of the two implicit SAS groups PUBLIC and SASUSERS.

With that, we can check off the introduction to users, groups and roles in SAS 9.4. Well, almost, we have to address one more thing: **Internal SAS Accounts**.

So far, we were talking about external users that are associated with a SAS identity (SASUSERS)– hopefully – because if not, they are public. Now, for good reasons, SAS implemented another group of users called *internal* SAS users.

SAS Internal Accounts

As the name implies, these user IDs are not in any way associated with an external account but are only known to SAS internally. If you don’t want to leave your house, you don’t have to care about anything that is happening outside. No street rules, no guidelines, no rules, no structure to follow; because you decided to stay inside and follow your own internal rules – which makes them internal rules, so to speak. Same with SAS internal users. Internal accounts do not really care about external users, policies, etc. They are solely focused on what is going on inside SAS. Think of it as a SAS account seclusion.

Internal SAS users are only known to SAS and are therefore authenticated internally only. You’ll recognize an internal user by the @-sign and the ending *saspw*. A good example is the unrestricted SAS admin account: **sasadm@saspw**. Because these accounts are internal and known only by SAS without any external “ties” or associations, they are easier to maintain. They are not affected by external password changes and there is minimal risk of security exposure.

Note: Even though internal user accounts are easier to maintain, do not create regular users as internal user. Only external users can establish SAS workspace server sessions, and those are needed if you want your users to use your SAS environment appropriately.

Another reason not to use internal accounts for general users is that users and groups should be subject to restrictions and rules that you put into place, including providing credentials for access. You want to know who does what, and what to monitor and report, Internal accounts should be used for SAS internal purposes only.

The following are some internal users that SAS needs and creates per by default:

sasadm@saspw

This is an unrestricted user, who can see and do everything in metadata. The unrestricted admin user is not subjected to permissions. It's the Superhero account amongst the admins.

sastrust@saspw

This is the SAS Trusted user, also known as the impersonator amongst the internal accounts. Instead of getting my newspaper on Saturday mornings, my dog gets it for me. I trust my dog Jasper that she will not shred, roll and then eat my newspaper, but bring it to me safely. I trust her and therefore she fulfills the "getting the newspaper tasks" for me. In other words, my dog acts as **sastrust**. It is something that doesn't affect or interest the outside world and is something that is internal to the Fischers' only.

sasevs@saspw

That is the account the SAS Environment Manager Server and its SAS Environment Manager Agents use to communicate with each other. The sasevs@saspw account could be compared with a video connection between two parties. I (SAS Env Manager Agent) Skype with my parents, Eri and Harald (Env Manager Servers), every Sunday (predictable). They are in Germany, I am in Cary, North Carolina, in the United States. The internet connection (sasevs@saspw) works well most of the time, which means, our communication and exchange of information works well. On some days, however, the connection is bad, and our information exchange is interrupted. You could say sasevs@saspw has the hiccups. When this hiccup happens between SAS Environment Manager Server and the SAS Environment Manager Agents, it means you would not get any information about your resources or any reports in the SAS Environment Manager tool.

webanon@saspw

This is an internal anonymous web account that you can use to give web clients access to your web services without them having to put in credentials. By using webanon you are opening it all up a bit; using the webanon user, the clients get access through this user. The webanon@saspw account could probably be compared to a *grandpa-said-its-ok-but-maybe-mom-said-no-but-I-cannot-remember* situation.

For documentation reference, see Appendix.

Why Internal Accounts?

Being internal and known by SAS only without any "ties" or associations externally makes it nice and easy to maintain. Internal accounts don't care about external password changes for example. Another great advantage is that you do not have to create dedicated external OS accounts for processes that might be purely internal to SAS.

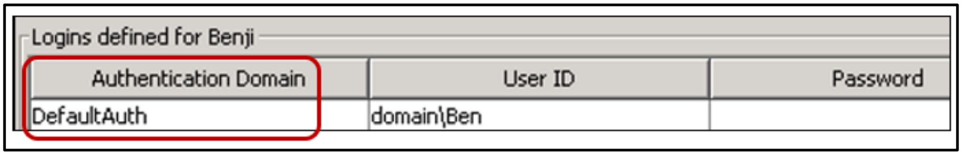
I hope that the SAS Users, Groups and Roles excursion is helpful to you as you go along with setting up, administering and maintaining your SAS environment. It is much needed knowledge for putting a good security structure into place and for understand who does what in your environment.

Before we conclude, we have to go over one more thing, to make the user and group management discussion complete. We have to talk about *Authentication Domains*.

Authentication Domains

When talking about authentication domains, I am referring to the field shown in Figure 4.6.

Figure 4.6: Authentication Domain field



Logins defined for Benji

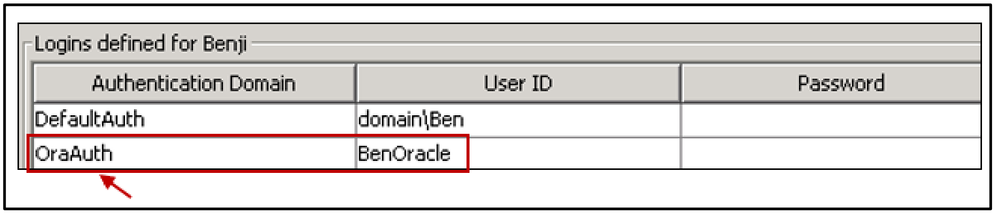
Authentication Domain	User ID	Password
DefaultAuth	domain\Ben	

As you know now, users in SAS metadata are associated with an OS/LDAP/AD/Database account. Going back to the example with Benji, you see one OS account in the properties of the metadata for user ID Benji. What you also see is a field for Authentication Domain. In this case, it shows *DefaultAuth*. An Authentication Domain comes into play for outbound logins. Outbound means, a user who is already authenticated by the metadata server makes a request for a workspace server, or, wants to access an external database. Everything that goes outside the initial (inbound) metadata server authentication. Inbound logins disregard and do not care about authentication domains. The DefaultAuth is only used to determine the user's metadata identity.

What if my user Benji does not only have to access data on the operating system, but, has to access data in Oracle as well? The logical thing to do would be – based on what we have learned so far – to probably create a new user ID in metadata for user Benji. But this time, we won't use the OS account, but will use the database user ID and password. Close, but not quite:

The beautiful thing about the users in metadata is that one metadata user ID can be associated with different external identities. Let's look at an example for user Benji. In addition to the DefaultAuth user ID, Benji also needs access to Oracle data. And for that reason, I simply add Benji's user ID for Oracle in his metadata server user ID. Now my metadata user ID looks like that in Figure 4.7.

Figure 4.7: Additional Authentication Domain



Logins defined for Benji

Authentication Domain	User ID	Password
DefaultAuth	domain\Ben	
OraAuth	BenOracle	

To make sure the metadata server understands what type of connection it must establish for this user, as soon as a user logs on with a user ID specified in the metadata ID, the metadata server looks for all logins in that user ID and checks if there is one that fits. For Benji, he accesses Oracle using the credentials BenOracle. The metadata server looks if there is such a user ID in Benji's metadata definition. Oracle is just one example. You can store database accounts, other OS accounts etc. in a user definition in metadata.

You could say: The reason why we use authentication domains is to simply allow separation of user IDs stored under one metadata identity. Think of an authentication domain as an address. If I visit my friends, I know exactly which direction I have to go. These directions could be called FriendAuth. If I go to work, I know how to get there – let's call it WorkAuth.

Authentication domains are used to even further qualify what user IDs are attached to a metadata server user.

Best Practice: Whenever you create an authentication domain to qualify an additional external user ID, make sure you use names that make sense. Naming an Authentication domain for Oracle or any other databases something like ABCAuth doesn't help if another admin has to administer, or , if you need to make changes later on. It is also important for troubleshooting purposes.

There is a great discussion about Authentication Domains on the SAS Administration and Deployment Community, which might be helpful to you. Paul Homes from Metacoda explains authentication domains in a great way, thinking of them as a tagging mechanism, "they are used to tag which credentials/logins can be used with which servers."

Summary

Let's summarize what we covered in this chapter:

- It is a best practice to create users and groups in metadata using the User Manager plug-in in SAS Management Console or SAS Environment Manager.
- Metadata user IDs are associated with operating system/LDAP/AD or database accounts.
- Metadata users are authenticated with the operating system, or LDAP/AD or internal before getting access to metadata.
- After successful authentication on the OS, users are considered members of the two implicit SAS groups PUBLIC and SASUSERS, sometimes Public-only.
- Internal SAS accounts are used for internal SAS use, and internal use only.
- Authentication domains are used to further associated a metadata identity with external logins.
- Before you start creating groups, make sure you have a good plan in place. Think about the teams or departments users are in, what data they might have to access and so forth.

What else?

There is a good amount of documentation out there. My recommendation is: Do not drive yourself nuts by feeling that you must actually read each and every single article, page and book about Users, Groups and Roles management. In fact, I would like to recommend that – for now– you focus on nothing but users, groups, roles and authentication domains. Everything user ultimately ends in authentication and authorization. We will discuss SAS security in a little bit. If you do want to read up on users, groups, roles and authentication domains, see the Appendix for link references.