



SAS[®] Viya[®] Platform: Identity Management

2026.06 - 2026.08

This document might apply to additional versions of the software. Open this document in [SAS Help Center](#) and click on the version in the banner to see all available versions.

Identity Management: Overview	3
SAS Viya Platform 2022.10 UID and GID Changes	3
Basics of Identity Management	3
LDAP Requirements	4
SCIM Requirements	5
Just-in-Time (JIT) Requirements	6
Identity Management: Guidelines	7
How to Configure SCIM	7
SCIM IAM Integration with the SAS Viya Platform	8
Configure SCIM Provisioning in the Identity Provider	11
Configure Private User Directory for SCIM	20
Single Sign-on	20
Deleting SCIM-Based Identities	20
Kerberos Hybrid Authentication	21
How to Configure LDAP	21
Configure an LDAP Identity Provider	21
Set Up Administrative Users	23
Sign In Using LDAP Credentials	24
How to Configure Just-in-Time Provisioning	24
Just-in-Time Provisioning Prerequisites	25
Provisioning Users	25
Synchronizing Group Membership	26
Attribute Mappings	26

Deactivating Inactive Users	27
Identity Management: How to (CLI)	27
Adding Members and Creating Groups	27
Bulk Loading of User and Group Identifiers	28
Listing Users and Groups	28
Listing Group Members and Memberships	29
Showing Details for Users and Groups	29
Updating Users and Groups	30
Deleting Users and Groups	30
Deleting SCIM-Based Users and Groups	31
Refreshing Users and Groups	32
Validating the Identities Configuration	32
Identities CLI Example	33
How to Secure the sasboot Password	34
Disable the Password Reset Feature	34
Reset the sasboot Password	35
How to Manage UIDs and GIDs with the Identities Service	35
UID & GID Considerations	35
Selecting Configuration Settings	36
How to Configure the sas.identities Settings	37
Default Behaviors of UID and GID	37
Managing UID and GID Values	38
How Are UID and GID Values Generated	38
How Values Are Loaded	40
Ignoring GID Values	41
Managing UID and GID Generated Conflicts	41
Changing UID and GID Values	42
Upgrading to SAS Viya Platform 2022.10 and Later	43
Deciding Which Combination of UID and GID Setting Values to Use	43
Bulk Load Identifiers for the Identities API	45
Bulk Load Identifiers for the Identities CLI	45
Identity Management: Reference	46
Initial Users — Sasboot Account	46
Custom Groups	46
Identities	52
Identities Service	55
sas.identities	55
sas.identities.providers.ldap	57
sas.identities.providers.ldap.connection	57
sas.identities.providers.ldap.group (Field Mappings)	59
sas.identities.providers.ldap.group (Additional Properties)	60
sas.identities.providers.ldap.user (Field Mappings)	61
sas.identities.providers.ldap.user (Other Properties)	63
Identity Management: Troubleshooting	64
General Troubleshooting Strategies	64
Cannot Access Cloud Analytic Services	64
Cannot View Users and Group Members	64
Cannot Access Esri Geographic Mapping Resources	65
Cannot Retrieve List of Users or Groups	65
Cannot Update User Information	65
Cannot Log In to the SAS Viya Platform	65

Membership in a sasapp LDAP, Custom, or Host Group Is Ignored	66
Resolving a 409 Conflict When Querying a User or Group Identifier	66
Cannot Perform Sync Due to Error	67
<i>Identity Management: Interfaces</i>	68

Identity Management: Overview

SAS Viya Platform 2022.10 UID and GID Changes

Starting with version 2022.10, updates have been made to the process for managing user and group identifier information. The updates include default assignments for new configuration properties for the identities service. The new default settings might contradict how your existing deployment is configured, resulting in users being unable to access content. For detailed information see [“How to Manage UIDs and GIDs with the Identities Service” on page 35](#).

Basics of Identity Management

Identity management includes setting up and maintaining the connection between the SAS Viya platform and your identity provider. Identity management also includes administrators managing the membership of custom groups and CAS roles.

The SAS Viya platform is configured by default to use Lightweight Directory Access Protocol (LDAP). As an alternative, you can change the configuration to use System for Cross-domain Identity Management (SCIM) or just-in-time (JIT) provisioning. LDAP must be disabled if SCIM or JIT is selected. If SCIM or JIT is used, either SAML or OpenID Connect must be used for authentication. The combined use of LDAP, SCIM, or JIT in a single, individual environment is not supported. See the following requirements for each provisioning method:

- [“LDAP Requirements” on page 4](#)
- [“SCIM Requirements” on page 5](#)
- [“Just-in-Time \(JIT\) Requirements” on page 6](#)

Using LDAP is the default strategy for pulling in identities from an external identity management system. It is oriented more toward on-premises deployments where the customer has an existing LDAP server. LDAP enables user name and password authentication with the SAS Viya platform, which includes direct access to CAS. LDAP supports UID and GID numbers through POSIX attributes. The SAS Viya platform provides different models for managing UID and GID identifiers that allow the Identities service to generate UID and GID information. For more information see [“How to Manage UIDs and GIDs with the Identities Service” on page 35](#).

Sites that want to run compute jobs and CAS actions with predefined user accounts and use file system access under those accounts should use LDAP with the POSIX attributes. LDAP should be

used if the SAS Viya platform is deployed in such a way that storage is shared between the SAS Viya platform and other systems.

The SCIM specification is widely used in cloud environments to share identity information with applications. It is a “push” model whereby the identity management system provisions user and group identities to downstream applications such as those on the SAS Viya platform, and regularly synchronizes to keep the information up-to-date. Since SCIM is a push model, it is largely configured from the identity management side. From the SAS Viya platform, you need to establish only a client identity (OAuth client) and give it authorization to call the SCIM APIs on the SAS Viya platform.

SAS identity management includes the following:

- giving users, groups, and custom groups access to SAS functionality
- managing the membership of custom groups and CAS roles

You can use SAS Environment Manager for most identity management tasks.

TIP Familiarize yourself with the [Predefined Custom Groups on page 47](#).

LDAP Requirements

You can choose LDAP if you are using local file stores and local user directories.

With LDAP, the SAS Viya platform talks directly to the LDAP server to pull user and group identities into the SAS Viya platform. Therefore, the SAS Viya platform deployments need to be able to directly connect to the chosen LDAP server, which is deployed to your chosen cloud provider. The LDAP connection contains user credentials, and as such encryption should always be used with an LDAP connection. The SAS Viya platform supports both LDAPS (LDAP over SSL) and startTLS on the LDAP connection.

You have a choice of LDAP providers—any LDAP v3 compliant server is supported such as Active Directory or [OpenLDAP](#). You can use Active Directory as your LDAP provider. By default, the SAS Viya platform pulls in identities from an external LDAP server.

- The SAS Viya platform must connect to the LDAP server using an anonymous bind or system account.
- Users and groups are queried from LDAP via a configured base distinguished name (DN). Customers with very large LDAP forests might wish to use multiple base DN's. This is possible only if users are authenticating through single sign-on.
- The SAS Viya platform must connect to the LDAP server using a system account or an anonymous bind. Users and groups in the directory are queried from a base DN that you specify. For enterprises with very large LDAP forests, you can specify multiple base DN's, but only if users are authenticating with single sign-on, such as SAML or OAuth.
- LDAP enables user name and password authentication to the SAS Viya platform, including direct access to CAS. Since sign-ins are critical to the security of a system, the SAS Viya platform must query LDAP on every sign-in attempt. Accounts are authenticated using “search and bind.”
- First, the system binds to the LDAP server using the configured system account or anonymous bind. Next, the server searches for the directory entry that matches the user name entered by the

user. Then, the server attempts to bind to LDAP using the DN of that directory entry and the password entered by the user. The user name used by the SAS Viya platform is then set using the canonical user name queried from the LDAP attributes. Therefore, the user name appears throughout the SAS Viya platform in the same sense as it appears in LDAP (not how it was entered by the user).

- Since sign-ins are critical to the security of a system, the SAS Viya platform must query LDAP on every sign-in attempt. This involves a single query for the user plus dozens or even more queries for each of the groups that the user is a member of. All of this places a very high demand on LDAP. As a result, performance is a big consideration.
- User and group membership information is cached in the SAS Viya platform identities service in order to service general queries. Only LDAP group memberships are cached. An example of a general query is where the SAS Viya platform identities service responds to request such as listing groups when constructing authorization rules. This cache can take several minutes to load at start-up, is refreshed periodically, and can be refreshed by the SAS administrator in SAS Environment Manager.
- For every sign-in attempt, the SAS Viya platform makes a single query for the user plus a query for each of the groups that the user is a member of. All of this places a very high demand on LDAP. As a result, performance is a big consideration and affects sign-in time. Connection pooling and other tuning parameters are available, so the tuning section of the administration guide should be reviewed carefully.
- LDAP supports UID, GID numbers, and secondary GID numbers through POSIX attributes. The SAS Viya platform uses these values if they exist. Sites that want to run compute jobs and CAS actions with predefined user accounts and use file system access under those accounts should use LDAP with the POSIX attributes.
- If the mail attribute is specified for LDAP accounts, it must have a non-null value that is unique for each user.

SCIM Requirements

You can choose SCIM if you are running in a cloud environment.

With SCIM, you configure the identity provider to push user and group identities into the SAS Viya platform. Therefore, the identity provider must be able to reach your SAS Viya platform deployment over the internet. A valid certificate from a third-party certificate authority (examples include, DigiCert, Verisign, and so on) is required.

It is important to realize the direction of the network connection for firewall and security rules. The connection is initiated by the identity provider since the identity provider is the SCIM client while the SAS Viya platform Identities service is the SCIM server.

The identity management system acts as a SCIM client.

Note: SCIM 2.0 (which SAS supports) is not backward compatible with SCIM 1.1 or the deprecated SCIM 1.0.

- The identity management system needs to reach the SCIM server or the SAS Viya platform deployment over the internet. The URL must begin with HTTPS and have a certificate that is signed by a public certificate authority.
- From the SAS Viya platform, administrators need to establish only a client identity (OAuth client) and give it authorization to call the SCIM APIs on the SAS Viya platform.
- When using SCIM, CASHostAccountRequired members can run under individual users' host accounts. The identities service generates unique UIDs and GIDs for users. For more information see [“CAS Server Authentication” in SAS Viya Platform Identity and Access Management: Fundamentals](#) and [“Identity Providers” in SAS Viya Platform Identity and Access Management: Fundamentals](#).
- The SAS Viya platform does not store passwords or accept passwords through SCIM. Therefore, user name and password authentication is not possible when using SCIM. Examples of scenarios that require user name and password authentication are:
 - service accounts for SAS Compute Server
 - shared accounts for CAS and SAS Compute Server
 - stored credentials for an individual identity for CAS and SAS Compute Server

The SAS Viya platform command-line interface (CLI) enables authentication using an authorization grant instead of a user name and password combination. With such a grant, the user authenticates from a web browser in order to obtain an authorization code for the CLI. See [“Obtain an Access Token” in SAS Viya Platform: Authentication](#) and [SAS Logon API](#) for more information.

- The standard SCIM core schema defined by RFC 7643 does not include UID and GID information. Therefore, the SAS Viya platform generates UID and GID numbers for users and these values do not correspond to any existing user accounts on the machines or file systems. This is an important issue to consider when planning for how files are used or shared outside of the SAS Viya platform deployment.
- UID and GID numbers are generated using a hashing algorithm. There is a small chance of a clash when using this algorithm. If a clash occurs, an override can be supplied. See [“Resolving a 409 Conflict When Querying a User or Group Identifier” on page 66](#).
- Private user directories are not directly supported but can be configured.
- The SAS Viya platform needs to retrieve user identities and group membership information from the customer's identity management system.
- Using SAML or OpenID Connect (OIDC) is required for SAS Logon Manager if you choose SCIM.

Just-in-Time (JIT) Requirements

Just-in-time (JIT) provisioning requires that users authenticate with either OIDC or SAML. JIT provisioning is triggered for only federated authentications through these two protocols. JIT provisioning is appropriate for use when your system can support a model where users must log in at least once before any SAS configuration (such as role assignments) can be applied to them. For more information see [“Just-in-Time Provisioning Prerequisites” on page 25](#).

Identity Management: Guidelines

The following basic guidelines contribute to simplicity and security:

- Familiarize yourself with the [predefined custom groups](#). Based on this information, determine which of your users to add to each predefined custom group.
- Limit membership in administrative roles and groups.
- Assume administrative group memberships or a CAS administrative role only when you need to perform tasks that require the extra permissions. In the CAS administrative role, you can relinquish the role when you are finished.
- If you delete a custom group, any custom rules that you created still exist. Manually delete such rules.
- As you plan your group structure, remember that you can use a group for either or both of these purposes:
 - To make shared resources available to multiple users. For example, you might use a group as the principal in an authorization rule or you might store shared credentials for a group.
 - As a parent to other groups. For example, if groupA and groupB should have identical access to multiple resources, you might assign both groups to a parent group, and grant access to the parent group.

Note: When you design a group structure, consider both clarity (Will others be able to interpret the structure?) and conciseness (Is the structure as minimal as possible?). In a complex authorization model, you might prioritize clarity, using more than the strict minimal number of groups and giving each group a name that describes its purpose.

How to Configure SCIM

This section provides detailed information for provisioning SCIM, including the following: registering an OAuth Client, configuring SCIM in Microsoft Entra ID, configuring SCIM in Okta, and configuring a private user directory for SCIM.

SCIM IAM Integration with the SAS Viya Platform

Overview

The SAS Viya platform is configured by default to use LDAP as an identity provider. As an alternative, you can change the configuration to use SCIM. Each environment must choose either SCIM or LDAP. The use of both LDAP and SCIM in a single, individual environment is not supported.

If SCIM is used, either SAML or OpenID Connect must be used for authentication. In addition, LDAP must be disabled if SCIM is selected. Because SCIM is a "push" model, the identity provider needs to access the SAS Viya platform deployment over the internet. It must use HTTPS and have a certificate that is signed by a public certificate authority.

For more information see ["SCIM Requirements" on page 5](#).

Note: Deployments in the cloud (for example, Azure AKS) can be used but require the firewall settings be updated in order to allow connections from the identity provider's IPs.

IMPORTANT Starting with 2022.1.3 and later, a new custom group named *SCIM* is automatically created. In addition, a new authorization rule is also created. This authorization rule gives members of the *SCIM* custom group access to the SCIM endpoints in the SAS Viya platform. As a result, an authorization rule no longer needs to be created manually.

Disable LDAP (Required)

IMPORTANT Using LDAP and SCIM concurrently in a single environment is not supported.

Note: The cache can be enabled or disabled without any effect on integrating with the identity provider.

- 1 From SAS Environment Manager, navigate to the configuration definitions for the Identities service. In the applications menu (☰) under **ADMINISTRATION**, select **Manage Environment**. In the vertical navigation bar, select .

In the list of services, locate the **Identities service** and select it.
- 2 Edit the **sas.identities.providers** configuration definition:
 - a In the configuration instances panel, select **sas.identities.providers**.
 - b Select Edit . Set the **ldap.enabled** flag to **off**.

- c Ensure that the **scim.enabled** flag is set to **on**.
 - d Click **Save**.
- 3 Restart the Identities service. For more information, see [“Managing a Specific Server or Service” in SAS Viya Platform: General Management of Servers and Services](#).

Register OAuth Client for SCIM Client

Overview

The identity provider needs an access token in order to access the APIs on the SAS Viya platform. A long-lived token is required to avoid routinely updating the identity provider configuration. Users must register an OAuth client for the identity provider and use the client credentials to generate an access token. You can perform this task using a manual client registration or a CLI client registration. Both of these methods register a new client ID "idp-client-id", use a secret "idp-client-secret" (select a different one for your setup), and set up the client ID to obtain tokens using client credentials with a suggested expiration of 63070000 seconds or 2 years. In addition, the token must be replaced before it expires. For more information see [“Register a Custom Application \(OAuth Client\)” in SAS Viya Platform: Authentication](#).

Note: After completing the registration, the client ID is now registered with the SCIM authority.

- CURL is a command line tool and a library that can be used to receive and send data between a client and a server or between any two machines connected over the internet. It supports a wide range of protocols such as HTTP, FTP, IMAP, LDAP, POP3, SMTP, and many more.
- The SAS Viya Platform CLI posts to a secure endpoint by default. See [“Command-Line Interface: Instructions when Downloading the CLI” in SAS Viya Platform: Using the Command-Line Interface](#). By doing so, it is requesting an OAuth token from SASLogon.

Manual Client Registration

To register a client manually, follow these steps:

- 1 Obtain a token to register the new client ID and secret.

The example below uses a password grant, but other options are available. Use an account that is a member of the SASAdministrators group:

```
INGRESS_URL=https://hostname.example.com; \
export BEARER_TOKEN=`curl -sk -X POST \
"${INGRESS_URL}/SASLogon/oauth/token" \
-u "sas.cli:" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d 'grant_type=password&username=myusername&password=mypassword' | awk -F: '{print $2}'|awk -F\" '{print $2}'`
```

Note: The initial portion of the curl command must be entered on one line. Multiple lines are used here to improve readability.

- 2 Verify that the command executed correctly:

```
echo "The registration access-token is: " ${BEARER_TOKEN}
```

Output should be similar to the following:

```
The registration access-token is: eyJhbGciOi-hz_-1eQhpfd_nZcVE9stGwqjJ4WzR2doHEd
```

- 3 Register the Identity provider client ID with client_credentials grant type.

```
curl -k -X POST "${INGRESS_URL}/SASLogon/oauth/clients" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $BEARER_TOKEN" \
-d '{
  "client_id": "idp-client-id",
  "client_secret": "idp-client-secret",
  "authorities": ["SCIM"],
  "authorized_grant_types": ["client_credentials"],
  "access_token_validity": 63070000
}'
```

The output should be similar to the following:

```
{
  "client_id": "idp-client-id",
  "authorized_grant_types": ["client_credentials"],
  "autoapprove": [],
  "access_token_validity": 63070000,
  "authorities": ["SCIM"],
  "lastModified": 1628126258430,
  "required_user_groups": []
}
```

For more information about registering a client manually see [“Register a Custom Application Manually” in SAS Viya Platform: Authentication](#).

CLI Client Registration

Clients can be registered with the SAS Viya platform command-line interface (CLI) . Use the CLI to register an OAuth client to enable access to SAS Viya platform APIs. The OAuth CLI plug-in is required. If you have not already downloaded the SAS Viya platform command-line interface, be sure to do so.

- 1 Log in to the SAS Viya platform CLI using an account that is a member of the SASAdministrators group. Here is an example:

```
sas-viya --profile my-admin-profile auth login
```

- 2 Register the client by running the OAuth plug-in to the SAS Viya platform CLI.

The following command registers a client to obtain tokens using a password grant. Run the command without the --grant-password option to display all of the grant parameters.

```
INGRESS_URL=https://hostname.example.com
sas-viya --sas-endpoint ${INGRESS_URL} oauth register-client \
--grant-password \
--grant-client-credentials \
--id idp-client-id --secret idp-client-secret \
--authorities "SCIM" \
--valid-for 63070000
```

Note: You should change INGRESS_URL to specify the host name for the endpoint in your environment, and secret should be a unique secret value for registering the client.

The output should be similar to the following:

```
Registering new client 'idp-client-id' with grant_type password...
idp-client-id has been registered as a client of the SAS environment at https://hostname.example.com.
Clients can use idp-client-id and idp-client-secret to obtain access tokens.
OK
```

Obtain an Access Token

Whichever method was used to register the client, administrators can obtain the access token using the client credentials.

```
ACCESS_TOKEN=`curl -skX POST "${INGRESS_URL}/SASLogon/oauth/token" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type=client_credentials" \
-u "idp-client-id:idp-client-secret"| awk -F: '{print $2}'|awk -F\" '{print $2}'; \
echo "The client access-token is: " ${ACCESS_TOKEN}; \
echo ${ACCESS_TOKEN} > ~/idtoken.txt
```

The output should be similar to the following:

```
The client access-token is: eyJhbGciOiJSUzI1NiIsImprSI6Imh0dHBzOi8vbG9jYWxob3N0L1NBU0xv...
```

Configure SCIM Provisioning in the Identity Provider

Initial Requirements for SCIM Provisioning

You must allow IP addresses that are used by the provisioning service to make SCIM requests to the SAS Viya platform deployment.

Configure SCIM Provisioning in Microsoft Entra ID

As an Azure Administrator, follow these steps to configure SCIM Provisioning:

- 1 Log on to the Azure portal and select **Microsoft Entra ID**.
- 2 On the left panel, select **Enterprise Applications** ⇒ **New Application** and enter **SAS Viya** in the search field. Click on **SAS Viya SSO** in the search results.
- 3 On the Create page, provide a name for your application.
- 4 Click **Create**. The new application is created.
- 5 Now you can add users and groups to your application. On the Overview page for the new application, under **Getting Started**, select **Assign users and groups**. On the Users and groups page, select **Add user/group**. On the Add Assignment page, you can add users and roles to your

application. Under **Users**, select **None Selected**. On the Users page that appears, select a user from the list. Click **Select**. On the Add Assignment page, the user is now listed. Click **Assign**. The user is now displayed on the Users and groups page.

Note: A common requirement is to sync only a subset of assigned users and groups configured with the enterprise application.

- 6 Now you can provision user accounts for your application. On the Overview page for the new application, under **Getting Started**, select **Provision User Accounts**. Then select **Get started**. On the Provisioning page that opens, select **Get Started**.
- 7 On the Provisioning page that opens, select **Automatic** from the **Provisioning Mode** menu. This expands the **Admin Credentials** section.
- 8 In the **Admin Credentials** section, complete the following steps:
 - a In the **Tenant URL** field, specify the base URL to the SAS Viya platform and append / `identities/scim/v2`.

Note: HTTPS is required. In other words, the ingress controller for your SAS Viya platform environment is configured for TLS and is using a certificate that is issued by one of the public certificate authorities that are supported by Microsoft Authentication Services.

- b In the **Secret Token** field, paste the Bearer token that was created when you followed the process for [“Obtain an Access Token” on page 11](#).
- c Click **Test Connection**.
If you configured the system correctly, you will receive a successful response.
- d Click **Save** on the Provisioning page. The **Mappings** and **Settings** sections become active.

IMPORTANT You must save the provisioning changes before proceeding.

- 9 Now you can define attribute mappings for your application. In the **Mappings** section, click **Provision Microsoft Entra ID Users**.

In the Attribute Mapping window that opens, complete the following steps:

- a Ensure that **Enabled** is set to **Yes**.
- b Attribute mappings define how attributes are synchronized between Microsoft Entra ID and customappsso. :

IMPORTANT Pay careful attention to the mapping for **userName**. Microsoft Entra ID maps this field to **userPrincipalName** by default. This must match what you use for the user name with single sign-on. If you set up single sign-on with SAML or OIDC and configure it to use email, change the **userName** mapping to use **mail** instead of **userPrincipalName**.

If you are using OIDC, another mapping option is to map **userName** to **objectId** in the SCIM configuration and use **oid** in the OIDC claims. For more information about ID token claims see [ID tokens in the Microsoft identity platform](#).

The mapping type should be set as an expression and not as a constant for the active field.

The following attributes are supported by the SAS Viya platform:

Table 1 *Attributes to Keep*

Microsoft Entra ID Attribute	customappsso Attribute
userPrincipalName	userName
Switch([IsSoftDeleted], , "False", "True", "True","False")	Active
displayName	displayName
jobTitle	title
mail	emails[type eq "work"].value
streetAddress	addresses[type eq "work"].street...
city	addresses[type eq "work"].locality
state	addresses[type eq "work"].region
postalCode	addresses[type eq "work"].posta...
country	addresses[type eq "work"].country
telephoneNumber	phoneNumbers[type eq "work"]...
mobile	phoneNumbers[type eq "mobile"]...
facsimileTelephoneNumber	phoneNumbers[type eq "fax"].va...
objectId	externalId

Click **Delete** to delete the other attributes.

- c Ensure that **Enabled** is set to **Yes**.
 - d Click **Save** and confirm with **Yes**.
- 10 Return to the main Provisioning page (from the top navigation or by exiting the Users provisioning window).

- 11 In the **Mappings** section of the Provisioning page, click **Provision Microsoft Entra ID Groups**.
In the Attribute Mapping window that opens, complete the following steps:
 - a Ensure that **Enabled** is set to **Yes**.
 - b Click **Delete** to delete all attributes except **displayName** and **members**, which are supported by the SAS Viya platform.
- 12 Click **Save** and then click **Yes** to confirm.
- 13 Now you can define settings for your application. Return to the main Provisioning page, and complete the **Settings** section:
 - a Select **Send an email notification when failure occurs**, and enter an email address in the **Notification email** field.
 - b For the **Scope** field, select **Sync only assigned users and groups**.
- 14 Set **Provisioning Status** to **On** and click **Save**.
- 15 Click on your Enterprise Application from the top navigation or click **X** from the top right to return to the main Provisioning page.
- 16 The **Start provisioning** action is inactive if synchronization has started. Likewise, the **Stop provisioning** is active.

The amount of time that Microsoft Entra ID takes to synchronize users and groups can vary from 20 minutes to several hours. This depends on the total number of users and groups. Once the synchronizing process is complete, you can see the status of how many users and groups have been synchronized and view the audit logs.

IMPORTANT You might have to delete all existing users from SAS if you are clearing state and restarting the synchronization from Microsoft Entra ID. In addition, you need to pickup users where only the case in the user name has changed.

Network Requirements for the Cloud Provider

Firewall Considerations for SCIM: From Microsoft Entra ID to SAS Viya Platform on Azure

Before you configure SCIM Provisioning in Microsoft Entra ID, be aware of these requirements:

- You must have administrator privileges in the Azure portal.
- You must allow IP addresses that are used by the Microsoft Entra ID provisioning service to make SCIM requests to the SAS Viya platform deployment. For a list of IP addresses for each service tag and region, see [Azure IP Ranges and Service Tags – Public Cloud](#). You can download and program these IP addresses into your firewall, as needed. The reserved IP address ranges for Microsoft Entra ID provisioning can be found under the **Microsoft Entra ID** section.

If deploying the SAS Viya platform in Azure and using VNET traffic from these addresses, the SAS Viya platform enables the network security group via a rule that enables the AzureActiveDirectory Service Tag.

- If you are using Azure WAF (Web Application Firewall,) be aware that Azure WAF blocks requests to the SAS SCIM endpoint as they are identified as malicious requests. To prevent this, you should create a custom allowlist rule for the SAS SCIM endpoint “/identities/scim/v2”.

Firewall Considerations for SCIM: From Microsoft Entra ID to SAS Viya Platform on Amazon Web Service

When configuring SCIM provisioning for Microsoft Entra ID Amazon Web Service, you must resolve limitations to use of AzureActiveDirectory Service Tag. AWS enforces a limit on the amount of inbound and outbound rules that you can have for an Amazon Virtual Private Cloud security group. The default limit is sixty inbound rules and sixty outbound rules for each Amazon VPC security group. This results in a total of 120 rules for each Amazon VPC security group. In contrast, the AzureActiveDirectory Service Tag has limit of 110 different IP ranges.

Your configuration should focus on the inbound traffic from Microsoft Entra ID via HTTPS. When you specify an IP range in the loadBalancerSourceRanges attribute, AWS creates three different inbound rules for that IP range:

- HTTP
- HTTPS
- Custom ICMP rule

As a result, you should increase the AzureActiveDirectory Service Tag limit to a minimum of 330 rules per Amazon VPC security group to properly accommodate Microsoft Entra ID. The following example shows a query of the Amazon VPC and the JSON file from Microsoft:

```

vpcname=myvpc

vpcid=$(aws ec2 describe-vpcs | jq -r --arg vpcname "$vpcname" '.Vpcs[] |
select(.Tags[]? | select(.Value==$vpcname)) | .VpcId')

groupid=$(aws ec2 create-security-group --group-name ukits-azuread --description
"Rules to allow SCIM provisioning from Microsoft Entra ID" --vpc-id $vpcid | jq -r
'.GroupId')

url=$(curl -s https://www.microsoft.com/en-us/download/confirmation.aspx?id=56519 |
grep -o '<a .*href=.*>' | sed -e 's/<a /\n<a /g' | sed -e 's/<a .*href=("["]*)// ' -e
's/("["]*).*$// ' -e '/^$/ d' | grep ServiceTags | head -1)

for iprange in $(curl -s ${url} | jq -r '.values[] | select(.id ==
"AzureActiveDirectory") | .properties.addressPrefixes[]' | sed '/:/:d')
do
    aws ec2 authorize-security-group-ingress --group-id $groupid --protocol tcp --port
443 --cidr $iprange
done

helm install viya4 \
  --version 3.36.0 \
  --namespace ingress-nginx \
  --create-namespace \
  --set controller.service.loadBalancerSourceRanges={"<my-network-cidr>,<natip>/32"} \

```

```
--set-string controller.service.annotations."service\.beta\.kubernetes\.io/aws-load
-balancer-extra-security-groups"="$groupid" \
  ingress-nginx/ingress-nginx
```

With this example, a new security group is created—one that is dedicated to receiving inbound traffic from Microsoft Entra ID. The code then iterates through each IP range in the Microsoft JSON file, and adds that IP range to the Security Group for HTTPS only. When the security group is in place that contains the needed IP ranges, you can create the ingress-nginx controller. Add the following annotation to the Ingress controller:

aws-load-balancer-extra-security-groups

This annotation takes the new security group and associates it with the Ingress load balancer. Prior to running this code, you should increase the rule limit for your Amazon VPC security group to a number that can accommodate the IP ranges specified in the `loadBalancerSourceRanges` attribute.

Firewall Considerations for SCIM: From Microsoft Entra ID to SAS Viya Platform on Google Cloud

When configuring SCIM provisioning for Microsoft Entra ID on Google Cloud Provider, you should consider that Google Cloud does not acknowledge the use of the AzureActiveDirectory Service Tag. This tag is implemented with the inbound rules of the Network Security Group that are associated with the Load Balancer. As a result, you need to identify the specific IP ranges that are used by the Microsoft Entra ID service.

To accomplish this, Microsoft provides a JSON file that lists every IP range that is used by the AzureActiveDirectory Service Tag (as well as every other Service Tag within Azure). This file is available here: [Azure IP Ranges and Service Tags – Public Cloud](#).

The Microsoft JSON file is updated frequently. Therefore, you should implement an automated check of this file to ensure that your inbound security rules are aligned with the latest file from Microsoft..

You can now customize the ingress-nginx controller. When installing the ingress-nginx controller, you need to provide a greater number of IP ranges in the `loadBalancerSourceRanges` attribute specification to account for every possible Microsoft Entra ID source IP range. You should update the `loadBalancerSourceRanges` attribute specification to query the JSON file from Microsoft and pull the address ranges for the AzureActiveDirectory Service Tag.

The following example shows the query of the JSON file from Microsoft:

```
url=$(curl -s https://www.microsoft.com/en-us/download/confirmation.aspx?id=56519 |
grep -o '<a .*href=.*>' | sed -e 's/<a /\n<a /g' | sed -e 's/<a .*href=[\"'\"'\"']//\' -e
's/[\"'\"'\"'].*$//\' -e
\'/^$/ d\' | grep ServiceTags | head -1)

helm install viya4 \
  --version 3.36.0 \
  --namespace ingress-nginx \
  --create-namespace \
  --set controller.service.loadBalancerSourceRanges="{<my-network-cidr>,<nat-ip>
/32,$(curl -s ${url} | jq -r '.values[] | select(.id == \"AzureActiveDirectory\") |
.properties.addressPrefixes[]' | sed '/:/d' | sed -z 's/\n/,/g' | sed 's/,*/$/
g'),<etc.>}" \
  ingress-nginx/ingress-nginx
```

Once these customizations are implemented, Google Cloud adds an inbound rule covering every possible Microsoft Entra ID source address. For more information about setting up Google Cloud for

SCIM see [Deploy 1Password SCIM Bridge on Google Cloud Platform](#) and “Reference Diagram” in *SAS Viya Platform Operations: Provisioning Resources*.

IMPORTANT For specific information about the ingress-nginx controller see “ingress-nginx Controller Vulnerability Mitigation” in *SAS Viya Platform: Deployment Guide*.

Firewall Considerations for SCIM: From Microsoft Entra ID to SAS Viya Platform On-Premises

If you are running SAS Viya platform on-premises, you can use the Microsoft Entra Provisioning Agent Wizard to configure and provision users to SCIM-enabled applications. For more information see [Microsoft Entra on-premises application provisioning to SCIM-enabled apps](#).

Configure SCIM Provisioning in Okta

Note: The SAS Viya platform supports SCIM 2.0. Okta supports an on-premises provisioning agent. However, it supports only SCIM version 1.

Note: You need to create another app in Okta to configure single sign-on. After this is completed, you can open a support case with Okta to merge your SCIM configuration with your single sign-on app, if desired.

To begin, log in to the Okta tenant that you are using.

- 1 Click **Admin**.
- 2 Select **Applications** from the menu and click **Add Application**.
- 3 Enter **SCIM** in the search field and a list of templates is displayed.
Look for **SCIM 2.0 Test App (OAuth Bearer Token)** in the results. You might have to select **See All Results**.
- 4 Select the template and click **Add**.
- 5 Edit the **Application label** field. Enter a name for the SAS Viya platform deployment.
- 6 Click **Done**.
- 7 Go to the **Provisioning** tab and select **Integration** from the **SETTINGS** menu. Click **Configure API Integration**. Select **Enable API Integration**.

Some fields are displayed.

In the **SCIM 2.0 Base URL** field, enter the full URL to the SAS Viya platform deployment plus, “/identities/scim/v2”. For example: <https://myserver.example.com/identities/scim/v2>.

In the **OAuth Bearer Token** field, enter the token obtained earlier using client credentials.

Note: Do not include "Authorization" or "Bearer" in this field. Okta already includes these tokens.

Verify Test API Credentials

- 1 Click the **Test API Credentials** to verify the connection.
- 2 Click **Save**.
- 3 Remain on the **Provisioning** tab and select **To App** from the **SETTINGS** menu on the left. Select the **Enable** check boxes next to **Create Users**, **Update User Attributes**, and **Deactivate Users**. The SAS Viya platform does not store user passwords. Therefore, leave **Sync Password** disabled. You need to set up single sign-on using SAML or OpenID Connect (OIDC) so that users can sign in.
- 4 Click **Save**.

Navigate to the **Sign On** tab. Click **Edit** in the Settings section. Scroll down to the **Credentials Details**. Here you can choose what value Okta sends to the SAS Viya platform for the user name field.

IMPORTANT The user name field needs to match the single sign-on configuration. By default it uses the Okta user name. Make sure **Create and update** is selected. Note that the SAS Viya platform does not handle changes to the user name easily since existing folders, files, and so on are not migrated under the new user name.

An additional consideration for the Username field is the use of an OpenID Connect (OIDC) for single sign-on. If you set up single sign-on with OIDC and configure it to use email, change the **userName** mapping to use *email* instead.

- 5 Click **Save**. All changes are saved.

Refer back to the **Provisioning** tab. Scroll to the **Attribute Mappings**. Notice it states that the Username is configured in the Sign-on settings. Modify the **Attribute Mappings** so that they match what is shown in the table below:

Note: The minimum attribute needed to create a SCIM user in the SAS Viya platform is the *userName*. The minimum attribute needed to create a SCIM group in the SAS Viya platform is the *displayName*.

1 **Table 2** *Attribute Mappings*

Attribute	Attribute Type	Value	Apply on
User name userName	Personal	Configured in Sign On settings	None
Given name givenName	Personal	user.firstName	Create
Family name	Personal	user.lastName	Create

familyName

Primary email	Personal	user.email	Create and update
Primary email type emailType	Personal	(user.email != null && user.email != "") ? 'work' : "	Create and update
Title	Personal	user.title	Create and update
Display name displayName	Personal	user.displayName	Create and update
Primary phone primaryPhone	Personal	user.primaryPhone	Create and update
Address type addressType	Personal	(user.streetAddresses != null && user.streetAddresses != "") ? 'work' : "	Create and update
Street address streetAddress	Personal	user.streetAddresses	Create and update
Locality	Personal	user.city	Create and update
locality	None	None	None
Region	Personal	user.state	Create and update
region			
Postal Code	Personal	user.zipCode	Create and update
postalCode			
Country code	Personal	user.countryCode	Create and update
country			

- On the **Assignments** tab you can now assign users to this application. This can be done by choosing **People** under **FILTERS** or selecting **Groups**. Okta sends these to the SAS Viya platform as you assign them. You should then be able to see them displayed in SAS Environment Manager.
- Go to **Push Groups** to select the groups that should be sent to the SAS Viya platform. Click on **Push Groups**. Choose either option to find groups. When you have selected all the groups that you prefer, click **Save**. Okta immediately sends them to SAS. You can view them in SAS Environment Manager.

- Provisioning logs can be viewed by clicking the **View Logs** link near the top of the screen above the menus.

Configure Private User Directory for SCIM

To configure the Private user Directory for SCIM, perform the following tasks:

- 1 From SAS Environment Manager, navigate to the configuration definitions for the Identities service. In the applications menu (≡) under **ADMINISTRATION**, select **Manage Environment**. In the vertical navigation bar, select .

In the list of services locate the **Identities service** and select it.

- 2 Update the `sas.identities` configuration instance:

- a In the **main configuration** area, enter `sas.identities` in the **configuration instance Filter** field.

- b Next to the `sas.identities` configuration instance, click .

- c In the **Edit sas.identities Configuration** window, scroll down to the **identifier.homeDirectoryPrefix** field.

In this field, add the absolute path to the directory that contains your users' private user directory. For example: `/home`.

- d Click **Save**.

The identities service returns a home directory in the specified location when you are using SCIM. Processes such as CAS can use this home directory.

Single Sign-on

Note: Single sign-on is required for authentication when using SCIM. For more information about single sign-on see [“Secondary Authentication \(OAuth\)” in SAS Viya Platform Identity and Access Management: Fundamentals](#).

Deleting SCIM-Based Identities

The SAS Viya Identities CLI provides the ability to delete SCIM-based users and groups.

For deleting SCIM-based group identities with the SAS Viya Identities CLI, the `delete-scim-groups` command is used. Here is an example:

```
sas-viya identities delete-scim-groups
```

For deleting SCIM-based user identities with the SAS Viya Identities CLI, the `delete-scim-users` command is used. Here is an example:

```
sas-viya identities delete-scim-users
```

For deleting inactive SCIM-based user identities with the SAS Viya Identities CLI, the `delete-scim-users --inactiveOnly` command is used. Here is an example:

```
sas-viya identities delete-scim-users --inactiveOnly=true
```

Note: You can also filter inactive SCIM-based user identities in SAS Environment Manager. Enabling this setting filters out the visibility of inactive user identities in SAS Environment Manager. It does not delete inactive user identities. Follow these steps to enable this property:

- 1 Log on to SAS Environment Manager as an administrator.
 - 2 Navigate to the Configuration page. From the **View** drop-down list, select **Definitions**. In the **Filter** field, enter `sas.identities.providers`.
 - 3 From the **Identities service** drop-down list on the right pane, click . Change the `scim.filterInactiveUsers` property to **On**.
 - 4 Click **Save**.
-

Kerberos Hybrid Authentication

Additional support has been added for Kerberos hybrid authentication scenarios. This support is applicable in situations where the environment is configured for single sign-on with OIDC and SAML and also requires Kerberos access to data sources. The changes that were made also provide enhanced support for constrained delegation. For more information, see [“Kerberos Hybrid Authentication” in SAS Viya Platform: Authentication](#).

How to Configure LDAP

Use SAS Environment Manager to set up user and group authentication.

Configure an LDAP Identity Provider

After completing the installation of the SAS Viya platform, you can configure an LDAP identity provider. LDAP is the default identity provider.

To configure the connection to your identity provider:

- 1 Sign in to SAS Environment Manager as `sasboot`.

Note: For more information, see [“Sign In as the sasboot User” in SAS Viya Platform: Deployment Guide](#).

- 2 Select the  from the side menu to open the Configuration page.
- 3 On the Configuration page, select **All Services** from the list and then select the **Identities service** from the list of services.
- 4 To configure connection properties in the **sas.identities.providers.ldap.connection** section, click **New Configuration**. In the New Configuration window, complete these steps:
 - a Specify values for the following required fields: **host**, **password**, **port**, **url**, and **userDN**. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.
 - b Click **Save**.

TIP In addition to SAS Environment Manager, you can also use the SAS Viya Platform CLI plug-in for LDAP.

- 5 To configure user properties in the **sas.identities.providers.ldap.user** section, click **New Configuration**:
 - a Specify a value for the **baseDN** required field. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

Note: When using the LDAP protocol, passwords are transmitted over the network as cleartext. To secure the deployment, SAS recommends that you configure encrypted LDAP connections. For more information, see [Overview: Encrypt LDAP Connections](#) in Encryption in SAS Viya Platform: Data in Motion.

For each property that represents a user-level field in SAS, specify a corresponding property in the LDAP server software.

TIP Consider specifying a custom filter to limit the user accounts that the SAS Viya platform returns from your LDAP server.

- b Click **Save**.
- 6 To configure group properties, in the **sas.identities.providers.ldap.group** section, click **New Configuration**. In the **New Configuration** window:
 - a Specify a value for the **baseDN** required field. For the remaining fields, review the default values and make changes, as necessary. The default values are appropriate for most sites.

For each property that represents a group-level field in SAS, specify a corresponding property in the LDAP server software.

TIP Consider specifying a custom filter to limit the accounts that the SAS Viya platform returns from your LDAP server.

- b Click **Save**.
- 7 To verify user and group information from the SAS Environment Manager side menu, select  to open the **Users** page.
On the **Users** page, select **Users** from the list in the toolbar. Your users should appear after a few minutes. It is not necessary to restart any servers or services. Then, select **Groups** from the list to display your groups.
- 8 Verify that user and group information is displayed correctly. If not, make any necessary changes to the identities service properties, then restart the Identities and SAS Logon Manager services.
 - a Get the names of the pods that contain SASLogon and identities:


```
kubectl get pods -n name-of-namespace | grep sas-logon
```

```
kubectl get pods -n name-of-namespace | grep identities
```
 - b Remove the SASLogon and identities pods:


```
kubectl -n name-of-namespace delete pod saslogon-pod-name
```

```
kubectl -n name-of-namespace delete pod identities-pod-name
```

Set Up Administrative Users

While you are signed on to SAS Environment Manager as the sasboot user, set up at least one SAS Administrator user as follows:

- 1 On the **Users** page in SAS Environment Manager, select **Custom Groups** from the list in the toolbar.
- 2 In the left pane, click **SAS Administrators**.
- 3 In the **Members** section of the right pane, click , and add one or more members to the group (including your own account, if applicable).
- 4 Sign out from SAS Environment Manager so that you are no longer signed in as the sasboot user.
- 5 If you added your own account to the SAS Administrators group, you can sign on again to SAS Environment Manager using that account.

Open SAS Environment Manager from a URL with the following format:

protocol://*name-of-ingress-host*:*port*/SASEnvironmentManager

protocol can be http or https, depending on how you have secured your deployment.

TIP Since SAS Administrators is an assumable group, the following prompt is displayed: Do you want to opt in to all of your assumable groups?. Select **Yes** if you want the extra permissions that are associated with the SAS Administrators group. The selection remains in effect until you sign out.

Sign In Using LDAP Credentials

Open SAS Environment Manager from a URL with the following format:

protocol://name-of-ingress-host:port/SASEnvironmentManager

protocol can be http or https, depending on how you have secured your deployment.

Sign in as one of the SAS Administrators that you set up in [“Set Up Administrative Users” on page 23](#).

How to Configure Just-in-Time Provisioning

Users who access SAS Viya are required to have a corresponding identity record in the SAS Identity Management system. Authorization decisions are then based on the groups that the user belongs to. How these identity records and group memberships are created depends on how the customer's environment is configured. SAS Viya supports [LDAP](#) and [SCIM](#) provisioning, and beginning with SAS Viya 2026.06, SAS Viya supports just-in-time (JIT) user management provisioning.

Just-in-time (JIT) provisioning is a user-management strategy that automatically creates, updates, and maintains user accounts in SAS Viya at the moment a user authenticates. This is in contrast to pre-populating user accounts through a scheduled synchronization or administrative import. Rather than requiring pre-configuration or ongoing synchronization, JIT provisioning works entirely reactively. Users and their group memberships are provisioned automatically. This happens the first time they log in through a federated identity provider (IdP) such as SAML or OIDC. This approach reduces administrative overhead and ensures that user records are always grounded in a real, active authentication event.

Just-in-time (JIT) provisioning is a suitable option for the following instances:

- Users are authenticated with a federated identity provider (SAML or OIDC) but cannot expose LDAP to SAS.
- Your system does not have the resources to configure and maintain a SCIM integration.
- Your identity provider does not support SCIM 2.0 outbound provisioning.
- Your system can support a model where users must log in at least once before any SAS configuration (such as role assignments) can be applied to them.

Just-in-time (JIT) provisioning is implemented with a Kubernetes transformer. See the README file that is included with the deployment assets that you download from your software order. The README file is provided in either of the following formats:

- `$deploy/examples/sas-logon-app/jit-provisioning/README.md` (for Markdown format)
- `$deploy/sas-bases/docs/configure_just-in-time_provisioning_of_users_and_groups_from_external_identity_providers.htm` (for HTML format)

Just-in-Time Provisioning Prerequisites

Just-in-time (JIT) provisioning requires that users authenticate with either OIDC or SAML. JIT provisioning is triggered for only federated authentications through these two protocols. You should configure SAS Logon Manager for OIDC or SAML authentication with your external identity provider. In addition, to enable group membership synchronization, the IdP provider configuration must include an `attributeMappings.external_groups` field that tells SAS Viya which OIDC token or SAML assertion carries the user's group list. This is configured under the following settings:

- `sas.logon.saml.providers` - used for SAML providers
- `sas.logon.oauth.providers` - used for OIDC providers

The value should be the name of the claim or attribute in the IdP OIDC token or SAML assertion that contains the list of groups the user belongs to. Without this mapping, SAS Viya cannot determine group membership and a group membership synchronization does not occur. The IdP must also be configured to include the claim or attribute in the OIDC token or SAML assertion that it generates and sends to SAS Viya.

IMPORTANT To implement JIT provisioning, SCIM must be enabled.

Provisioning Users

When a user authenticates with a federated IdP for the first time, SAS Logon validates and processes the OIDC ID token or SAML assertion from the IdP. If JIT is enabled, it checks whether a corresponding user already exists in the SAS Identities service. If no record exists, a new user is provisioned automatically. If a record exists, but is marked inactive, it is reactivated. (For example: deactivated users are marked inactive).

Synchronizing Group Membership

During authentication, SAS updates the user's group memberships based on the groups provided by the identity provider (IdP). A user's group memberships are resolved during authentication. This process involves the following:

- SAS Logon retrieves the authoritative group list from the SAS Identities service.
- External groups that do not exist in SAS are provisioned and missing group memberships for the user are added.
- Groups that the user is no longer a member of (stale memberships) are also removed.

IMPORTANT Group membership synchronization during JIT login applies only to external groups sourced from the IdP (those groups mapped via `attributeMappings.external_groups`). Custom groups created and managed directly in the SAS Identities Service are not affected. Those memberships continue to be managed through the SAS Identities Service.

Attribute Mappings

Just-in-time (JIT) provisioning relies on additional attributes from the identity provider. These attributes are passed in the form of OIDC claims in the ID token, or attributes in the SAML assertion.

- *email*
- *given_name*
- *family_name*
- *phone_number*
- *external_groups*

The external identity provider should be configured to include these attributes, either as optional claims in the ID token for OIDC, or as optional attributes in the assertion for SAML. The name of the `external_groups` attribute can be configured with the `attributeMappings.external_groups` field in the configuration definition. For more information see [“Just-in-Time Provisioning Prerequisites” on page 25](#).

Note: For JIT, LDAP, and SCIM, email is an important value that must be unique on the system.

Deactivating Inactive Users

As part of the SAS Identity Management system, orphaned user accounts are identified and prevented from accumulating. A daily scheduled job checks for active users who have not logged in within a configurable threshold. This threshold can be overridden by setting the property `sas.auth.jit.max.inactive` to the desired number of days (The default is ninety days). These users are deactivated in both SAS Logon (prevents new tokens from being issued) and the SAS Identities Service.

Identity Management: How to (CLI)

The following examples assume that you have already signed in to the SAS Viya platform at the command line. See [“Command-Line Interface: Instructions when Downloading the CLI” in SAS Viya Platform: Using the Command-Line Interface](#).

Adding Members and Creating Groups

You can add members and groups to your application.

add-member

Adds a new member to a group.

Example: Add *user1* to the group that has the ID *NWSupport*.

Options for the *add-member* command include:

- **group-id** - Indicates the ID of the group that is being updated.
- **group-member-id** - Indicates the ID of the group being added. Use this option when adding one group to another group.
- **user-member-id** - Indicates the ID of the user that is being added to the group.

create-group

Creates a new group.

Example: Create a group with the name *USMarketing*. The group ID is *USMKT* and the description is *US Marketing Group*.

Options for the *create-group* command include:

- **description** - Indicates the description of the new group.
- **id** - Indicates the ID of the new group.
- **name** - Indicates name of the new group

Note: If a group is created with no name, the specified ID is used for the name.

Bulk Loading of User and Group Identifiers

You can bulk load user and group identifiers to your application from a JSON or CSV file.

bulkload-group-identifiers

Bulk loads group identifiers to your application.

Example:

```
sas-viya identities bulkload-group-identifiers --file path-to-file
```

bulkload-user-identifiers

Bulk loads user identifiers to your application.

Example:

```
sas-viya identities bulkload-user-identifiers --file path-to-file
```

The *file* option specifies the file name of a JSON (.json) or CSV (.csv) file that contains the identifiers that you are loading.

For more information see [“Bulk Load Identifiers for the Identities CLI” on page 45](#).

Listing Users and Groups

You can list users and groups for an application.

list-groups

Lists the groups that are defined in the SAS system.

Example: List the groups that have *Pune* in their group name

```
./sas-viya identities list-groups --name-contains Pune
```

The *list-groups* command lists only 50 items at a time by default. To list more than 50 items, you can use the `--limit` option.

Example:

```
./sas-viya identities list-groups --name-contains NorthAmerica --limit 100
```

list-users

Lists the users that are defined in the SAS system.

Example: List users whose name contains *Smith*.

```
sas-viya identities list-username contains "Smith"
```

Example: List users who belong to the ldap provider.

```
sas-viya identities list-users --providerId "ldap"
```

Common options for the *list-groups* and *list-users* commands are:

- **description-contains** - Returns any identity that contains the specified string.
- **id** - Returns the identity with the specified ID.
- **limit "50"** - Specifies the maximum number of identities to return. This command lists only 50 items at a time by default. Increase or decrease this number if needed.
- **name** - Returns any identity that has the specified name.
- **name-contains** - Returns any identity whose name contains the specified string.

Listing Group Members and Memberships

You can list group members and group memberships for an application.

list-members

Lists the members of the group.

Example: List the members of the Product Support group that has the group ID *ProdSupport*. Limit the number of members listed to 75.

```
sas-viya identities list-members -group-id ProdSupport -limit 75
```

list-memberships

Lists the memberships in which the user or group belongs.

Example: List the groups that the user ID *jesmith* belongs to.

```
sas-viya identities list-memberships -user-id jesmith
```

Common options for the *list-members* and *list-memberships* commands are:

- **group-id** - Specifies the ID of the group for whom you want to list members or memberships.
- **limit "50"** - Specifies the maximum number of group memberships to return. This command lists only 50 items at a time by default. Increase or decrease this number if needed.
- **sort-by "id"** - Specifies the primary sort criteria for group memberships. By default, the sort is in ascending order.

Showing Details for Users and Groups

You can show details for users and groups for an application.

show-group

Shows details about a particular group.

Example: Show details for the marketing group *NWMkt*.

```
sas-viya identities show-group --id NWMkt
```

show-user

Shows details about a particular user.

Example: Show details for the user *user1*.

```
sas-viya identities show-user --id user1
```

Common options for the *show group* and *show-user* commands are:

- **id** - Indicates the ID of the group or user that you want to show
- **show-advanced** - Shows advanced properties for the group.

Updating Users and Groups

You can update information for users and groups for an application.

update-group

Updates information about an existing group.

Example: Change the name of the group *Logistics* to *CorpLogistics*. Change the description to *CorporateLogistics*. The group ID is *LOG*.

```
sas-viya identities update-group --id LOG --name CorpLogistics --description CorporateLogistics
```

Options for the *update-group* command include:

- **description** - Specifies a new description for the group.
- **gid "0"** - Specifies the group identifier (GID) number to set or update for the group. This value must be greater than 0.
- **id** - Specifies the ID of the group that is being updated.
- **name** - Specifies a new name for the group.

update-user

Updates information about an existing user. This function supports updating the user identifier (UID) and group identifier (GID) values.

Example: Change the GID number for user Mark Smith. The user ID is *MRSMI*.

```
sas-viya identities update-user --id MRSMI --gid 300004
```

Options for the *update-user* command include:

- **gid "0"** - Specifies the group identifier (GID) number to set or update for the user. This value must be greater than 0.
- **id** - Specifies the ID of the user that is being updated.
- **uid "0"** - Specifies the user identifier (UID) number to set or update for the user. This value must be greater than 0.

For more information about working with group identifiers(GID) and user identifiers (UID) see [“How to Manage UIDs and GIDs with the Identities Service” on page 35](#)

Deleting Users and Groups

You can delete users, members, groups, and identifiers from an application.

delete-group

Deletes the specified group.

Example: Delete the group Logistics. The ID of the group is *LOG*.

```
sas-viya identities delete-group --id LOG
```

delete-group-identifier

Deletes the identifier for the specified group. This command is primarily used if your system is configured to have custom identifiers and you want to change the value for a group.

Example: Delete the identifier for the group Warehouse Inventory. The ID of the group is *WI*.

```
sas-viya identities delete-group-identifier --id WI
```

delete-user-identifier

Deletes the identifier for the specified user. This command is primarily used if your system is configured to have custom identifiers and you want to change the value for a user.

Example: Delete the identifier for user1. The ID of the user is *user1*.

```
sas-viya identities delete-user-identifier --id user1
```

remove-member

Deletes the specified member from an existing group. This can be a user or another group.

Example: Remove user1 from the marketing group. The ID of the user is *user1*. The group ID is *marketing-group*.

```
sas-viya identities remove-member --group-id marketing-group --user-member-id user1
```

Options for the *remove-member* command include:

- **group-id** - Specifies the ID of the group that is being updated.
- **group-member-id** - Specifies the ID of the group that is being removed from an existing group.
- **user-member-id** - Specifies the ID of the user that is being removed from an existing group.

Deleting SCIM-Based Users and Groups

You can delete all SCIM users or all SCIM groups from an application. Optionally, you can also delete inactive SCIM-based users .

delete-scim-groups

Deletes all SCIM-based groups from the system. Here is an example:

```
sas-viya identities delete-scim-groups
```

delete-scim-users

Deletes all SCIM-based users from the system. As an option, you can delete all inactive SCIM-based users with the **inActiveOnly** option. Here are examples:

```
sas-viya identities delete-scim-users
```

```
sas-viya identities delete-scim-users --inActiveOnly=true
```

For more information see “ [Deleting SCIM-Based Identities](#)” on page 20.

Refreshing Users and Groups

You can refresh the users and groups in the internal cache.

refresh-cache

Example:

```
./sas-viya identities refresh-cache
```

Validating the Identities Configuration

You can validate the identities configuration of the SAS system. This command enables you to pre-validate your LDAP configuration properties. You can validate your LDAP connection information, and you can validate the LDAP user and group properties, verifying that they return identities.

validate-config

Examples: Validate the LDAP configuration values. You can use any of the following commands to validate the LDAP configuration values:

- Specify the LDAP configuration values via the command line.

```
sas-viya identities validate-config --ldap-url "url" --ldap-userDN "userDN"
--ldap-password password --ldap-group-baseDN "group-baseDN"
--ldap-user-baseDN "user-baseDN"
```

- Specify a file that contains the LDAP configuration values.

```
sas-viya identities validate-config --file path-to-file
```

- Specify a file that contains the LDAP configuration values. Use the `ldap-userDN` and `ldap-password` options to specify a different LDAP user and password to override these values from the file. If a command contains LDAP options and the file option, the LDAP option values override the values in the file.

```
sas-viya identities validate-config --ldap-userDN "userDN" --ldap-password password
--file path-to-file
```

Options for the *validate-config* command include:

- **file, f** - Specifies the file name of a JSON or YML file to validate the configuration with.
- **ldap-anonymous** - Enables anonymous binding to the LDAP server.
- **ldap-group-baseDN** - Specifies the LDAP baseDN for groups.
- **ldap-password** - Specifies the LDAP login password.
- **ldap-url** - Specifies the url of the LDAP server to validate the configuration against.
- **ldap-user-baseDN** - Specifies the LDAP baseDN for users.
- **ldap-userDN** - Specifies the LDAP login user name.

Consider the following key points about the *validate-config* command:

- This command is not used for setting up LDAP.
- You can use this command only on a secure (TLS-enabled) environment.
- You can include the LDAP configuration values in a separate file by using the `--file` option. The file that you specify can be a JSON or YML type.
- If LDAP is your identity provider, these points apply:
 - The password that is used to connect to the LDAP server must be Base64-encoded. If you specify the password with the `--ldap-password` option, the CLI encodes it for you. If you specify the password in the JSON or YML file, you must encode the password explicitly.
 - Here is a typical file of the type YML that is used for validating your LDAP configuration values:

Note: Enter the URL definition on a single line. Multiple lines are used here to improve readability.

```
sas.identities.providers.ldap.connection:
  host: ldap.sun.com
  password: xxxxxxxxxxxx=
  port: 389
  url: ldap://${sas.identities.providers.ldap.connection.host}:
    ${sas.identities.providers.ldap.connection.port}
  userDN: cn=admin,dc=sun,dc=com
sas.identities.providers.ldap.group:
  baseDN: DC=SUN,DC=com
  accountId: cn
  objectClass: groupOfUniqueNames
  objectFilter: (objectClass=groupOfUniqueNames)
sas.identities.providers.ldap.user:
  baseDN: DC=SUN,DC=com
  accountId: uid
  objectClass: inetOrgPerson
  objectFilter: (objectClass=inetOrgPerson)
```

Identities CLI Example

In the following example, the EastCoast sales group for a company is being split into the NorthEast and SouthEast sales groups. New groups are created and existing members are moved to the new groups.

Show the details for the EastCoast sales group.

```
sas-viya identities show-group --id eastcoast
```

List the members of the EastCoast sales group.

```
sas-viya identities list-members -group-id eastcoast
```

Create a new group named NorthEastSales.

```
sas-viya identities create-group --name NorthEastSales --id NESales --description "North East Sales group"
```

Create a new group named SouthEastSales.

```
sas-viya identities create-group --name SouthEastSales --id SESales --description "South East Sales group"
```

Add existing members to the new NorthEastSales group. Add user1, user2, and user3 to the group.

```
sas-viya identities add-member --group-id NESales --user-member-id user1
```

```
sas-viya identities add-member --group-id NESales --user-member-id user2
```

```
sas-viya identities add-member --group-id NESales --user-member-id user3
```

Add existing members to the new SouthEastSales group. Add user4, user5, and user6 to the group.

```
sas-viya identities add-member --group-id SESales --user-member-id user4
```

```
sas-viya identities add-member --group-id SESales --user-member-id user5
```

```
sas-viya identities add-member --group-id SESales --user-member-id user6
```

You can now optionally delete the EastCoast sales group.

```
sas-viya identities delete-group --id eastcoast
```

Refresh the users and groups in the internal cache.

```
./sas-viya identities refresh-cache
```

How to Secure the sasboot Password

Disable the Password Reset Feature

When you are finished setting up SCIM or LDAP and the initial administrative users, you should reset the password for the sasboot user. For additional security, you can then disable the password reset feature. This action prevents password reset links from being written to the log each time the SASLogon service is restarted.

- 1 Sign in to SAS Environment Manager as an administrative user and select  from the side menu to open the Configuration page.
- 2 On the Configuration page, select **Definitions** from the drop-down list.
- 3 In the left pane, select **sas.logon.initial**. Then click **New Configuration** at the top of the right pane. If a definition already exists, you can click  to edit the existing definition.
- 4 In the New sas.logon.initial Configuration window or the Edit sas.logon.initial Configuration window, set **reset.enabled** to **off**.
- 5 Click **Save**.
- 6 You can now restart the SASLogon service. To do so, you must delete the associated pod. The pod automatically restarts after it has been deleted. For more information, see [“Managing a Specific Server or Service” in SAS Viya Platform: General Management of Servers and Services](#).

Reset the sasboot Password

After you disable this feature, you can still change the sasboot password if the existing password is known. Enter the URL for the SAS Viya platform with the path `/SASLogon/change_password`. If you are already signed in as another user, first sign out and then sign back in as sasboot using the current password. You can then complete the steps to change the password.

How to Manage UID and GID with the Identities Service

Starting with 2022.10, updates have been made to the process for managing user and group identifier information. The SAS Viya platform now provides different models for managing UID and GID identifiers. With the new update, and as the SAS Administrator, you can allow the Identities service to generate UID and GID information. Alternatively, you can provide your own UID and GID information, which would be loaded to the Identities service. In addition, you can load UID and GID information either from LDAP or manually.

UID & GID Considerations

In your SAS Viya platform environment, the Launcher Service, SAS Cloud Analytic Services, and the SAS/CONNECT Spawner can all launch processes as end users. Here are some considerations for each of these:

- The Launcher Service always obtains the user attribute information from the Identities service or the values registered to a custom application, which includes the UID and possibly the GID information. This is used for the pod that is then launched.
- For SAS Cloud Analytic Services, the default option is `CASCLLOUDNATIVE=1`. For Host Launched sessions, the user attribute information is returned from the Identities service or the values registered to a custom application.

For more information about the `CASCLLOUDNATIVE` option see [“The CASHostAccountRequired Custom Group” on page 49](#).

- For the SAS/CONNECT Spawner, the default option is `SASCLLOUDNATIVE=1`. The user attribute information is queried from the Identities service or the values registered to a custom application.

For more information about the `SASCLLOUDNATIVE` option see [“SASCLLOUDNATIVE” in SAS Viya Platform: Programming Run-Time Servers](#).

If users interact with the file system from the SAS Compute Server, SAS Cloud Analytic Services host-launched sessions, or from SAS/CONNECT sessions, the UID and GID that is used is crucial to the implementation of file system permissions.

Selecting Configuration Settings

Below are new configuration settings for the Identities service:

- `sas.identities.identifier.generateUids` = **true**/false
- `sas.identities.identifier.generateGids` = true/**false**
- `sas.identities.identifier.disableGids` = true/**false**

The following table shows expected behaviors for the setting `sas.identities.identifier.generateUids`:

Table 3 *Generate UID Behavior*

Generate UIDs	Expected Behavior
true (default)	This setting always supplies a generated UID that is returned when the user's identifier endpoint is returned. Any attempt to load UID values through the Identities service endpoint returns an error. Users with existing, user-provided UIDs (loaded to identities when generate was set to false or persisted in LDAP) are preempted by the generation of UIDs. This overrides any user-provided UIDs.
false	This is a user-provided model. Custom UID values can be loaded through the Identities service endpoint or persisted in the LDAP store. The Identities service endpoints return user-provided UID values. Loaded values should take precedence over LDAP. If a user-provided value is not found, then nothing is returned.

The following table shows expected behavior for the two settings used to control GID values; `sas.identities.identifier.disableGids` and `sas.identities.identifier.generateGids`:

Table 4 *Disable GID Behavior*

Disable GIDs	Generate GIDs	Expected Behavior
false (default)	true	This setting always supplies a generated GID. This is returned when either identifier endpoint is called (users or groups). Any attempt to load GID values through the Identities service endpoint returns an error.

Disable GIDs	Generate GIDs	Expected Behavior
		Groups with existing user-provided GIDs (loaded to identities when generate was set to false or persisted in LDAP) are preempted by the generation of GIDs. This overrides any user-provided GIDs.
false (default)	false (default)	This is a user-provided model. Custom GID values can be loaded through the Identities service endpoint or persisted in the LDAP store. The Identities service endpoints return user provided GID values. Loaded values should take precedence over LDAP. If a user-provided value is not found, then nothing is returned.
true	Not applicable	GID functionality is disabled. GIDs are not returned by either Identities service endpoints in any context.

Note: For both `sas.identities.identifier.disableGids` and `sas.identities.identifier.generateGids`, the default setting is **false**.

As a SAS Administrator, you choose to either provide your own values or to generate values based on these configuration settings. Any previous configuration settings are not used.

How to Configure the sas.identities Settings

The `sas.identities.identifier` configuration options are part of the `sas.identities` group of settings. If you do not want to use the default values, a SAS Administrator can set the values. Here are the different ways that you can set the values. You can use either of these methods to set the configuration options:

- 1 Use SAS Environment Manager to manually configure the settings. The Identities service automatically retrieves the configuration change after a few of minutes.
- 2 Use the SAS Viya platform CLI to either create or patch the configuration by providing a JSON configuration file.

Default Behaviors of UID and GID

By default, with SAS Viya platform 2022.10 and later releases, the following occurs:

- The Identities service always provides a generated UID, even if your LDAP Provider has a valid POSIX UID attribute.

- The Identities service does not allow you, as a SAS Administrator, to load UID values through the Identities service endpoint.
- The Identities service allows you, as a SAS Administrator, to provide GID values from your LDAP Provider.
- The Identities service allows you, as a SAS Administrator, to load GID values through the Identities service endpoint.
- The Identities service does not generate a GID value if either of the following are true:
 - GID values are not provided by your LDAP Provider.
 - Nothing is returned when the identities service is loaded.

The default behavior is to generate UID values and to provide GID values.

Managing UID and GID Values

As shown in [Default Behaviors of UID and GID on page 37](#), UID values are not treated the same as the GID values. In the default case, the assumption is that you want the Identities service to generate UID values, but if you do not provide GID values, then no GID values are returned. As a SAS Administrator, you can choose one of the following:

- generate UID values and provide your own GID values
- provide your own UID values and generate GID values
- provide your own UID & GID values
- generate both UID & GID values
- ignore GID values altogether

How Are UID and GID Values Generated

When the Identities service is configured to generate values for GID or UID, the same approach is used. Values are generated when the Identifier endpoint is called. The following table lists the identifier endpoints.

Table 5 *Identifier Endpoints*

Identifier	Identifier Endpoint
USER	<code>https://viya.customer.com/identities/users/{{USERNAME}}/identifier</code>
GROUP	<code>https://viya.customer.com/identities/groups/{{GROUPID}}/identifier.</code>

To generate values, the Identities service takes the identity type (either USER or GROUP) along with the `accountId` value and generates an SHA hash of the string. This ensures that a user with the same

accountId value as a group generates a different hash value. The hash value is then mapped onto a numerical space to generate either the UID or GID value.

If the UID is generated for a user, their primary GID (returned as just gid from the Identifier endpoint) is always set to the same value. Only when generating GID values for groups are these values populated as secondaryGid for the end-user's Identifier endpoint. For example, the following output shows the user Identifier endpoint when generating User and GID values:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <identityIdentifier version="1"> <links>
<link href="/identities/users/sasadm/identifier" method="GET"
rel="self" type="application/vnd.sas.identity.identifier" uri="/identities/users/sasadm/identifier"/>
<link href="/identities/users/sasadm" method="GET"
rel="user" type="application/vnd.sas.identity.user" uri="/identities/users/sasadm"/>
  <link href="/identities/users/sasadm/identifier" method="PUT"
rel="update" type="application/vnd.sas.identity.identifier" uri="/identities/users/sasadm/identifier"/>
</links>
<gid>1681596511</gid>
<id>sasadm</id>
<secondaryGids>
  <secondaryGid>456501765</secondaryGid>
  <secondaryGid>436234771</secondaryGid>
  <secondaryGid>1713176570</secondaryGid>
  <secondaryGid>1180074129</secondaryGid>
  <secondaryGid>1729685794</secondaryGid>
  <secondaryGid>940932761</secondaryGid>
  <secondaryGid>173401048</secondaryGid>
  <secondaryGid>114936103</secondaryGid>
  <secondaryGid>661725102</secondaryGid>
</secondaryGids>
<uid>1681596511</uid>
</identityIdentifier>
```

Whereas, the following output shows the user Identifier endpoint if you are generating only UID values.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <identityIdentifier version="1"> <links>
<link href="/identities/users/sasadm/identifier" method="GET" rel="self"
type="application/vnd.sas.identity.identifier" uri="/identities/users/sasadm/identifier"/>
<link href="/identities/users/sasadm" method="GET"
rel="user" type="application/vnd.sas.identity.user" uri="/identities/users/sasadm"/>
<link href="/identities/users/sasadm/identifier" method="PUT" rel="update"
type="application/vnd.sas.identity.identifier" uri="/identities/users/sasadm/identifier"/>
</links>
<gid>1681596511</gid>
<id>sasadm</id>
<uid>1681596511</uid>
</identityIdentifier>
```

The same output is provided by the user Identifier endpoint if you set sas.identities.identifier.disableGids to true as well. Only the primary GID is returned.

IMPORTANT Starting in 2023.06, the hashing algorithm used by the identities service to generate UID and GID values was updated. Values that have already been generated remain the same. However, newly generated UIDs and GIDs are likely to result in different values. These new values could affect users' ability to access file-system resources if permissions

have been set based on the old values. Permissions issues are particularly likely to occur if multiple environments must access the same file-system resources.

Contact SAS Technical Support if you need assistance adjusting file system permissions as a result of this change.

How Values Are Loaded

When, as a SAS Administrator, you decide to provide your own values, you have a choice in how to provide them. If you have an LDAP Provider with a POSIX schema, the Identities service can retrieve the UID or the GID values from your LDAP Provider along with the other user and group attributes. Alternatively, you can load UID or GID information directly into the Identities service. This can be done with either the SAS Viya platform CLI or the Identities service REST API.

With the SAS Viya Platform CLI, you can update individual users:

```
/opt/sas/viya/home/bin/sas-viya identities update-user
Command requires the following flags: id
```

NAME:

`sas-viya identities update-user` - Updates information about an existing user. Currently only supports updating the user identifier (uid/gid) values.

USAGE:

```
sas-viya identities update-user [command options] [arguments...]
```

OPTIONS:

`--gid "0"` Specifies the gid number to set or update for the user. This value must be greater than 0.

`--id` Specifies the ID of the user that is being updated.

`--uid "0"` Specifies the uid number to set or update for the user. This value must be greater than 0.

Command requires the following flags: id

With the SAS Viya Platform CLI you can also update individual groups:

```
/opt/sas/viya/home/bin/sas-viya identities update-group
Command requires the following flags: id
```

NAME:

`sas-viya identities update-group` - Updates information about an existing group.

USAGE:

```
sas-viya identities update-group [command options] [arguments...]
```

OPTIONS:

`--description` Specifies a new description for the group.

`--gid "0"` Specifies the gid number to update for the group. This value must be greater than 0.

`--id` Specifies the ID of the group that is being updated.

`--name` Specifies a new name for the group.

Command requires the following flags: `id`

In addition, the [Python Tools for SAS Viya Platform](#) also offers an option to set the POSIX user attributes for users and groups.

Beginning with release version 1.18.11 of the identities plug-in, you can also bulk load either users or groups by providing either a JSON or CSV input file. The current available version of the identities plug-in is 1.18.9.

Ignoring GID Values

Setting `sas.identities.identifier.disableGids` to `true` prevents the Identities service from completely retrieving GID information. This applies regardless of the method of providing GID information, that is, whether you provide your own values or you generate GID values.

If your file system security model is primarily based on UID, you can, as a SAS Administrator, set `sas.identities.identifier.disableGids` to `true`. This means that the Identities service responds slightly faster since there is no internal processing for GID values. With this setting, there is no LDAP queries for GID information or any internal database queries for GID information.

Managing UID and GID Generated Conflicts

The UID- and GID-generated values are calculated by taking an SHA hash and then mapping the resulting hash value onto a numerical space. The SHA hash is unique. However, it is possible that when mapping onto the numerical space, two different hash values could result in the same numerical value. A conflict or clash occurs when the Identifier endpoint is called for a user or group and the value that is generated is the same as an existing generated value. If such a clash does occur, the Identities service retries generating the value, moving slightly the numerical space the hash is mapped onto. This prevents clashes in generate mode from producing an exception, and you are guaranteed to correctly generate a value.

However, if a conflict occurs and you have multiple environments (where you expect the generated values to be the same) you still might have a conflict issue. Since the UID or GID values are generated when the Identifier endpoint is called (for example: logging in to SAS Data and AI Studio), you might find that your end users have triggered the generation in a different order.

Note: Prior to 2026.06, SAS Data and AI Studio was known as SAS Studio.

The following is an example scenario:

- If user A and user B both generate the same UID, then whichever one is queried first gets the original UID value. The second user gets the newly generated alternative value .
- On the first system, User A is queried first and is given the value **1234**. User B is queried second and is given the value **9876**. These are their identifiers.
- Then, if on a different system, user B's identifier is queried first and is given the value **1234**. User A's identifier is queried second and is given the newly generated, alternative value.

One way to resolve this issue is to use the SAS Viya Platform CLI to remove the two identifier values and then regenerate them. The SAS Viya Platform CLI provides the following command to delete user identifier information:

```
/opt/sas/viya/home/bin/sas-viya identities delete-user-identifier
```

Command requires the following flags: id

NAME:

`sas-viya identities delete-user-identifier` - Deletes the identifier for the specified user.

USAGE:

`sas-viya identities delete-user-identifier [command options] [arguments...]`

OPTIONS:

`--id` Specifies the ID of the user whose identifier is being deleted.

Command requires the following flags: id

The SAS Viya Platform CLI also provides the following command to delete group identifier information:

```
/opt/sas/viya/home/bin/sas-viya identities delete-group-identifier
```

Command requires the following flags: id

NAME:

`sas-viya identities delete-group-identifier` - Deletes the identifier for the specified group.

USAGE:

`sas-viya identities delete-group-identifier [command options] [arguments...]`

OPTIONS:

`--id` Specifies the ID of the group whose identifier is being deleted.

Command requires the following flags: id

Alternatively, you could provide your own UID and GID values.

Changing UID and GID Values

Starting with 2022.10, you can change the values for either UIDs or GIDs. If you change the **true** or **false** value of one of the settings, the Identities service automatically corrects the value the next time the Identifier endpoint is called.

For example, if you are generating both UID and GID values, but then decide that you need to provide the GID values, you can change the setting `sas.identities.identifier.generateGids` to **false**. You can then provide those GID values either from your LDAP with POSIX attributes or by loading them. The next time the Identifier endpoint is called (for example; by a user logging into SAS Data and AI Studio), the new value is used.

Although the Identities service now uses the value that you want, separately, you must ensure that any file system or other external content is correctly updated.

Note: Prior to 2026.06, SAS Data and AI Studio was known as SAS Studio.

Upgrading to SAS Viya Platform 2022.10 and Later

If you are upgrading to 2022.10 or later from an earlier release, then the new settings for managing the user and group identifier information apply after you upgrade. Any previous configuration settings that enable the generation of UID or GID information no-longer apply and have no effect.

If, as a SAS Administrator, you do not choose a model for managing the user and group identifier information, then the default settings apply. In addition, the Identities service automatically updates and corrects your user and group identifier information. As a result, previous values for UID or GID used by the SAS Viya platform might be lost.

Therefore, before you upgrade to 2022.10 or later you should understand how your existing environment is configured and how this maps to the new model for the provision of UID and GID information. Otherwise, your end users will not be able to access content after the upgrade.

For example, if you leveraged LDAP with POSIX attributes to provide your UID values in a previous release, after upgrading the new default model would generate UID values. As soon as the Identifier endpoint is called for a given user, their UID values are replaced. This happens as soon as your end users log in to SAS Data and AI Studio (known as SAS Studio prior to 2026.06) because the Launcher service calls the Identifier endpoint.

Deciding Which Combination of UID and GID Setting Values to Use

Here are some different scenarios for setting the UID and GID identifier values.

Scenario: SCIM Environment with No File System Access

In this case UID and GID values are not important. You would use generated UID and primary GID values. However, you do not need secondary GID values. As a result, you would set the options to the following:

- `sas.identities.identifier.generateUids = true.`
- `sas.identities.identifier.generateGids = false.`
- `sas.identities.identifier.disableGids = true.`

Scenario: POSIX Attributes in LDAP, Using a Shared File System

In this case you have already set user attributes in your LDAP system. Therefore, you want to ensure that the SAS Viya platform uses these LDAP POSIX attributes. As a result, you would set the options to the following:

- `sas.identities.identifier.generateUids = false`
- `sas.identities.identifier.generateGids = false`
- `sas.identities.identifier.disableGids = false`

Scenario: Environment with Shared File Systems, Leveraging an Existing Group-Based Security Model

In this case you are not concerned with the UID values, but the secondary GID values are vital to your file system security model. This file system security model already exists, and the SAS Viya platform must use the same GID values as your existing systems. Therefore, you would allow the SAS Viya platform to generate the UID values and you would provide your own GID values. As a result, you would set the options to the following:

- `sas.identities.identifier.generateUids = true`
- `sas.identities.identifier.generateGids = false`
- `sas.identities.identifier.disableGids = false`

If you have an LDAP with the POSIX group attributes, you could use this. Alternatively, if you do not have the GID values in LDAP you could instead load these into SAS Viya Platform.

Scenario: Environment with Shared File Systems, Leveraging a New Group-Based Security Model

In this case you are not concerned with the UID values. Rather, the secondary GID values are vital to your file system security model. However, this is a new group-based security model and it applies only to the SAS Viya platform. You are able to use the GID values generated by the SAS Viya platform to build this new file system security mode. Therefore, you can allow the SAS Viya platform to generate both the UID & GID values. As a result, you set the options to the following:

- `sas.identities.identifier.generateUids = true`
- `sas.identities.identifier.generateGids = true`
- `sas.identities.identifier.disableGids = false`

You can then call the group identifier endpoint to discover the generated GID values to use in building your new security model.

Bulk Load Identifiers for the Identities API

The Identities API now provides endpoints that enable you to bulk load UID and GID values for identities that are stored in the `extended_attributes` Postgres table. The input can contain both UID and GID values, and the API handles JSON input for the REST call.

The identity must exist in the system prior to calling an endpoint to add identifier values to them. If the user or group is not found, the entry is logged in the identities service log and skipped. The system continues processing the entries. Here are the required input values:

- *IdentityType* - user or group identity.
- *IdentityId* - the ID of the user or group whose identifier values you are updating or adding.
- *GID Value* - the primary GID for a user or the GID value of the group. If this value is not specified for a user, the endpoint currently assigns the UID value as the primary gid value for a user.
- *UID Value* - the UID value for the user.

Here is an example JSON input:

```
[
  { "type" : "user", "id" : "user1", "uid" : 200001, "gid" : 300001 },
  { "type" : "user", "id" : "user2", "uid" : 200002, "gid" : 300002 },
  { "type" : "user", "id" : "user3", "uid" : 200003 }
]
```

Note: You are not allowed to add secondary GID values to a user as they are not stored in the Postgres table.

Bulk Load Identifiers for the Identities CLI

The SAS Viya Identities CLI provides bulk loading of user and group identifiers. The input can be either a JSON file or a CSV file for either user or group identifiers.

For bulk-loading group identifiers with the SAS Viya Identities CLI, the **bulkload-group-identifiers** command is used. You can bulk load group identifiers from a JSON or CSV file. You must specify the file name of a JSON or CSV file to load identifiers with. Here is an example:

```
sas-viya identities bulkload-group-identifiers --file path-to-file.
```

For bulk-loading user identifiers with the SAS Viya Identities CLI, the **bulkload-user-identifiers** command is used. You can bulk load user identifiers from a JSON or CSV file. You must specify the file name of a JSON or CSV file to load identifiers with.

```
sas-viya identities bulkload-user-identifiers --file path-to-file.
```

Here is a JSON file example:

```
[
  { "type" : "user", "id" : "user1", "uid" : 200001, "gid" : 300001 },
  { "type" : "user", "id" : "user2", "uid" : 200002, "gid" : 300002 },
  { "type" : "user", "id" : "user3", "uid" : 200003 }
]
```

Note: The last line of this example shows that if the GID value for the user is missing, it uses the UID value (200003) as the GID value also.

Here is a CSV file example:

```
user,user1,200001,300001
user,user2,200002,300002
user,user3,200003,
```

Note: The last line of this example shows that if the GID value for the user is missing, it uses the UID value (200003) as the GID value also.

For more information about the SAS Viya Identities CLI see [“Identity Management: How to \(CLI\)” on page 27](#).

Identity Management: Reference

Initial Users — Sasboot Account

For more information, see [“Sign In as the sasboot User” in SAS Viya Platform: Deployment Guide](#).

Custom Groups

What Is a Custom Group?

A custom group is a group that exists in the SAS Viya platform but not in your identity provider. These groups are persistent in a SAS database.

Your deployment includes a set of [Predefined Custom Groups on page 47](#). You can also create your own custom groups. This feature is useful for creating new groups of SAS users if you do not want to (or do not have permission to) create groups in your identity provider.

Assumable Custom Groups

The SAS Administrators group is a predefined custom group. This group is *assumable*. When a user in this group signs in to the SAS Viya platform, a prompt appears asking Do you want to opt in to all of your assumable groups?

If the user selects **Yes**, the user gets the extra permissions that are associated with the assumable group. If the user selects **No**, the user does not get the extra permissions. The selection remains in effect until the user signs out.

As a best practice, users should select **Yes** only when they need to perform tasks that require the extra permissions.

Predefined Custom Groups

Certain custom groups are provided with your deployment. These groups provide an easy way to give users and groups access to the appropriate data, content, or functionality.

- The custom groups effectively implement a role within the SAS Viya platform. The members of the custom groups have access to privileges that are associated with the role. Search on the **Rules** page by the group name to see all privileges that are associated with the role.
- The predefined groups below are a part of a deployment that contains SAS Visual Analytics, SAS Visual Statistics, and SAS Visual Data Mining and Machine Learning. Some products and solutions have additional predefined groups. See the documentation for these products and solutions for information about other predefined groups.

The custom groups are as follows:

Table 6 Custom Groups

Custom Group	Description
Application Administrators	Can access selected administrative functions within applications. This group is used with SAS applications such as SAS Theme Designer, SAS Visual Analytics, and SAS Model Manager. Is not an assumable group. Has no initial members.
Data Agent Administrators	Data Agent Administrators group. SAS Data Agent administrators create data services, DSNs, and domains. For more information see “Data Access Controls” in Cloud Data Exchange for the SAS Viya Platform: Administrator’s Guide.
Data Agent Power Users	Data Agent Power Users group. SAS Data Agent Power users can add credentials for themselves to a domain, connect with DSNs and services, and process data.

Custom Group	Description
	For more information see “Data Access Controls” in <i>Cloud Data Exchange for the SAS Viya Platform: Administrator’s Guide</i>.
Esri Users	Can access Esri systems for geo map access. To access Esri premium services, the SAS user must be a member of the Esri Users custom group. In addition, the user must have an account on the Esri server. Is not an assumable group. Has no initial members. For more information see “Esri Premium Services Needs Valid Credentials” in <i>SAS Viya Platform: External Credentials</i> and “Configure Single Sign-On Access to Esri Services” in <i>SAS Viya Platform: Authentication</i>. Note: Esri requires that organizations pay for tokens to use the Esri geographic mapping services. You can add a user or group of users to the Esri Users group to control who has access to these tokens. Therefore, you can control the cost of using Esri geographic services.
Launcher Super Users	Super User group for SAS Launcher. The Launcher application is a command-line utility that is used with SAS Real-Time Decision Manager. For more information see SAS Real-Time Decision Manager.
SAS Administrators	Have access to the following: ■ all functionality that is controllable through authorization rules. ■ all folders and all objects that the folders contain (for example, plans and reports). Is an assumable group. Members can assume the CAS Superuser role. Members of this group can create domains, groups, and add credentials for other users on their behalf. Members can also manage access to resources and secure content. Note: Access to data (caslibs) is not included. For example, users in this group can create, run, and view reports only if they have explicitly been granted access to the underlying data.
SAS Score Users	Provide full access to score definitions and score execution APIs. This group is used with SAS Model Manager. SAS Model Manager configures the SAS Score Users group automatically. For more information see “Granting Access to Test Results” in <i>SAS Model Manager: Administrator’s Guide</i>.
SCIM Clients	Provides access to the SCIM endpoints within SAS Viya. For more information about SCIM see “SCIM Requirements” on page 5.
Service Account Users for Batch	Users allowed to store an OAuth 2.0 service account credential for Batch servers.

Custom Group	Description
	For more information see “Configure Batch Servers to Run under a Shared Account” in SAS Viya Platform: Server Contexts.
Service Account Users for Compute	Users allowed to store an OAuth 2.0 service account credential for Compute servers. For more information see “Configure Compute Servers to Run under a Shared Account” in SAS Viya Platform: Server Contexts.
Service Account Users for Schedule	Users allowed to store an OAuth 2.0 service account credential for others to use in Jobs and Flows. For more information see: “What Is a Group-Managed Service Account” in SAS Viya Platform: Jobs and Flows and “Configure a Group Managed Service Account for Jobs and Flows” in SAS Viya Platform: External Credentials.
Subject Matter Experts	Subject Matter Experts for Information Catalog. For more information see “Enable Users or Custom Groups to Update Review Status and Item Descriptions” in SAS Data Governance: Administrator's Guide.
Visual Investigator Administrators	Users with elevated access to SAS Visual Investigator. For more information see SAS Visual Investigator.
Visual Investigator Auditors	Users authorized to run SAS Visual Investigator audit reports. For more information see SAS Visual Investigator.
Visual Investigator Users	Users authorized to use SAS Visual Investigator. For more information see SAS Visual Investigator.
Visual Scenario Administrator Designers	Users with elevated access to SAS Visual Scenario Designer. For more information see SAS Visual Scenario Designer.

Note: An additional custom group is predefined, but not created. If you create a group with ID: *CASHostAccountRequired*, an overlay must also be applied to the base kustomization.yaml file. By default, CAS sessions run using the **cas** account. For more information, see [CASHostAccountRequired custom group on page 49.](#)

The CASHostAccountRequired Custom Group

The CASHostAccountRequired custom group is predefined, but not created. If you create a group with ID: *CASHostAccountRequired*, an overlay must also be applied to the base kustomization.yaml file. For more information see [“Enable Host Launch” in SAS Viya Platform: Deployment Guide](#).

Here are additional details about the CASHostAccountRequired custom group:

- By default CAS sets the environment variable CASCLOUDNATIVE to 1. This means that for members of the CASHostAccountRequired custom group, their UID or GID is queried from the SAS Viya platform Identities service and not the operating system.
- If a user is a member of the CASHostAccountRequired custom group, but has no host account, then SAS Environment Manager cannot access information about the CAS Server. You might observe that from SAS Environment Manager, the CAS server appears to be down even though it is not. No libraries or tables are displayed.
- When you modify the membership of this group, the users that have been added or removed must log off from their sessions before the changes can take effect.
- All data associated with a user's caslib is stored in a specific CASUSER path location on the CAS controller. Note that CASCLOUDNATIVE=1 is the default. In SCIM mode (CASCLOUDNATIVE is defined). There are two ways to override the location: CASMAKEHOMEDIR and setting the consul key value.

Table 7 CASUSER Path Location

User Scenario	CASUSER Path Location	Session Information
User starts CAS sessions from visual interfaces (includes all SAS Viya platform interfaces and Base SAS or SAS Programming Runtime Environment [SPRE] sessions), and user is not a member of the CASHostAccountRequired custom group. This is the default behavior.	<code>/cas/data/caslibs/casuserlibraries/USERNAME</code>	Sessions run under the CAS service account user (sas). The directory and all files within it are owned by the SAS user.
User starts CAS sessions from visual interfaces (includes all SAS Viya platform interfaces and Base SAS or SPRE sessions), and user is a member of the CASHostAccountRequired custom group.	Note: The use of \$HOME depends on where your home directory eventually resides. <code>\$HOME/casuser</code>	Sessions run under the user's host account.
User starts CAS sessions from Base SAS or SPRE, regardless of whether the user is a member of the CASHostAccountRequired custom group. Connections from Base SAS and SPRE require use of an OAuth token.	Note: This applies only when CASCLOUDNATIVE is off. <code>\$HOME/casuser</code>	Base SAS and SPRE sessions always run under the user's host account and use the <code>\$HOME/casuser</code> CASUSER path location. Sessions run under the CAS server user (cas). The directory and all files

User Scenario	CASUSER Path Location	Session Information
		within it are owned by the cas user.

If a user is added to the CASHostAccountRequired custom group, the next CAS session that is started for that user uses `$HOME/casuser` for the CASUSER path location. If this user needs to maintain access to any user files that were stored in the default CASUSER path location:

- A Kubernetes administrator must copy the files from the default CASUSER path location `/cas/data/caslibs/casuserlibraries/USERNAME` to `$HOME/casuser`.

When in SCIM mode, if the Identities service does not supply a home directory, the user's CASUSER ends up in `/cas/data/caslibs/casuserlibraries/$USER/casuser`.

Note: Users need to determine where `$HOME` is and the destination they need copy to.

Note: If a user is removed from the CASHostAccountRequired group, files should be copied in the opposite direction.

- An administrator must adjust the permissions on the files so that the user can access them. Prior to the user being added to the CASHostAccountRequired custom group, the cas user owned the files and performed all read or write actions.

In a SCIM environment, if the administrator inadvertently omits creating the correct UID or GID values from the outset, they should check the UID or GID of the casuser directory and use these values to determine file ownership.

Note: It is recommended that users use the ownership of the casuser directory to determine this value.

Note: As an alternative to assigning users to the CASHostAccountRequired custom group, you can set the environment variable `env.CASALLHOSTACCOUNTS`. This variable can be set when you want CAS sessions for all users in the environment to run under the host identity of the requesting user. For more information see [“env.CASALLHOSTACCOUNTS” in SAS Viya Platform: SAS Cloud Analytic Services](#).

See Also

[“Manage Custom Groups” in SAS Environment Manager: User's Guide](#)

Identities

Identity Filtering

When configuring the connection to your identity provider, you should specify a filter to limit the identities that the SAS Viya platform returns. For example, you can create a filter to exclude identities whose accounts are disabled or expired or to exclude objects that represent computer resources rather than actual users or groups. You can modify this filter at any time.

If you have a large number of users, using a filter can improve performance and reduce memory requirements. In addition, user management tasks can be performed more efficiently if only relevant identities are listed in SAS Environment Manager.

A default filter is provided for sites that use Active Directory. If you use another identity provider such as openLDAP, then you might need to modify the default filter. For more information about the default filter, see [“Identities Service” on page 55](#).

Identity Caching

Identity caching is available for enhanced performance. Search requests go to the cache thereby reducing the number of direct requests to the identity provider. You can configure the cache refresh interval and enable or disable the cache. The cache is enabled by default.

Identities Synchronization

Information about identities is available in SAS Environment Manager and is synchronized with the SAS Infrastructure Data Server (PostgreSQL) periodically. The amount of time between each synchronization is determined by a configuration option, which is set to 12 hours by default.

If you want to manually synchronize the identities at any time, you can [reload the identities](#). Note that reloading the set of users and groups can take several minutes to complete.

LDAP Group Membership Cache

In some SAS Viya platform installations, it is possible that users are members of multiple LDAP groups. When the sas-identities-service attempts to authenticate a user, it cycles through all membership groups. This can add a time delay to a user login.

The Identities service has an in-memory, group membership cache. Entries in the group membership cache are purged after an amount of time (six hours by default). If an entry is not found in the group membership cache, or it has expired, the Identities service attempts to retrieve the group again from the LDAP server. As queries are made for group memberships, they are added to the group membership cache. Any subsequent calls for those group memberships are found in the group membership cache. And they are returned without calling LDAP again. If not found, the Identities service queries LDAP for the group membership.

Starting in release 2026.02, you can enable preloading of an LDAP group membership cache. The preloading of the cache can result in faster user authentication in certain conditions. See the table [Preloading Group Membership Cache on page 53](#) for examples.

Some considerations for the LDAP group membership cache include the following:

- Preloading of the LDAP group membership cache is disabled by default.
- If preloading is not enabled, the refresh interval uses the existing LDAP group membership cache interval.
- The LDAP group membership cache size can be configured. By default, it is 1000.
- When initially loading the LDAP group membership cache, it defaults to a refresh interval that matches the identity cache.
- Loading the LDAP group membership cache typically does not impact loading the identity cache.

Here are the identities properties that enable pre-loading of the LDAP group membership cache:

`sas.identities.providers.ldap.membershipCache.preload`

enables you to pre-load the LDAP group membership cache to allow faster membership queries. Pre-loading this cache helps address potential slowness when querying memberships from LDAP. Preloading applies the LDAP groups objectFilter to collect the groups from LDAP and load them into the membership cache. Set this environment variable to true on the identities container to enable pre-loading. This environment variable is set to false by default.

`sas.identities.providers.ldap.membershipCache.size`

enables you to set the LDAP group membership cache size. This property defaults to 1000. This cache size is customizable with an upper limit of 3000. If a limit higher than 3000 is provided, then the max limit is set to 3000.

Here are some scenarios for preloading the group membership cache:

Table 8 *Preloading Group Membership Cache*

Scenario	Result
The group membership cache is not preloaded.	When a member's groups are loaded, and the group is not in the cache, it is added.
The group membership cache is preloaded. The cache size is greater than 0 and less than the limit.	When a member's groups are loaded, and the group is not in the cache, it is not added.
The group membership cache is preloaded. The cache size is equal to 0 or equal to the limit.	When a member's groups are loaded, and the group is not in the cache, it is added.

Required SCIM Attribute Mappings to Microsoft Entra ID

Attribute mappings define how attributes are synchronized between Microsoft Entra ID and the SAS Viya platform (customappsso). To configure SCIM provisioning with Microsoft Entra ID, specific attributes are mapped during your configuration on the Azure portal.

The following attributes are required for SCIM provisioning with the SAS Viya platform:

Table 9 Required SCIM Attribute Mappings to Microsoft Entra ID

Microsoft Entra ID Attribute	customappsso Attribute
userPrincipalName	userName
Switch ([IsSoftDeleted], , "False", "True", "True", "False")	Active
objectId	externalId

The following attributes are not required but could be useful for your SCIM configuration:

Table 10 SCIM Attribute Mappings to Microsoft Entra ID

Microsoft Entra ID Attribute	customappsso Attribute
displayName	displayName
mail	emails[type eq "work"].value

For more information see [“Configure SCIM Provisioning in Microsoft Entra ID”](#) on page 11.

Identities Properties That Require LDAP Attributes

In order for the Identities service to function properly, the following identities properties must be assigned valid LDAP attributes:

sas.identities.providers.ldap.connection

host

port

password

userDN

sas.identities.providers.ldap.group

accountId

baseDN'

distinguishedName

member

memberOf

objectClass

objectFilter

```

searchFilter
sas.identities.providers.ldap.user
  accountId
  baseDN1
  distinguishedName
  memberOf
  objectClass
  objectFilter
  searchFilter

```

¹ If you are in an Active Directory environment, this is the only required LDAP attribute for this group. Default values are provided for the other properties.

Note: The default values are valid for most implementations of Microsoft Active Directory. For other LDAP providers, such as OpenLDAP, you must provide different values for some fields. For all LDAP providers, it is recommended that you review the values for the required LDAP attributes to ensure that they are configured correctly.

Identities Service

The Identities service retrieves information about identities (users or groups) from your identity provider. It also enables the creation and management of custom groups. For general information about configuring services, see [SAS Viya Platform: Configuration Properties](#).

Here are the configuration properties for the Identities service:

sas.identities

The set of properties that are used to configure global values for the Identities service.

```
cache.cacheRefreshInterval
```

The refresh interval for the identities cache.

Note: Do not specify **cache.cacheRefreshInterval** below 20 minutes. Doing so might have a significant impact on your overall system, especially on the LDAP and SAS Infrastructure Data (PostgreSQL) servers.

Use the following conventions to specify the unit of time for the refresh interval:

- d - refers to days (for example, 6d).
- h - refers to hours (for example, 6h).

- m - refers to minutes (for example, 6m).
- s - refers to seconds (for example, 6s).
- ms - refers to milliseconds (for example, 6ms).

cache.enabled

Enables identities information to be cached. Caching is enabled by default.

cache.providerPageLimit

The number of identities to process in a given request when loading the cache. The default value is 1000.

defaultProvider

The default provider. The default value is `local`. (For this release, SAS recommends that you do not change this value.)

identifier.disableGids

Disables the lookup and generation of GIDs for users and groups. The default value is `false`.

identifier.disallowedUids

A comma delimited list of identifier UID values to disallow. The default value is 1001.

identifier.generateGids

Generates GIDs for users and groups. The default value is `false`.

identifier.generateUids

Generates UIDs for users. The default value is `true`.

identifier.homeDirectoryPrefix

Specify the absolute path to a directory that can contain the private user directories. If you set a value of `/home` and a user *andrea* logs in, the directory `/home/andrea` is the home directory for that user. Processes such as CAS use this home directory.

Note: Mounting of home directories can be disabled by leaving the value of this property blank.

SAS.ExternalIdentity

Provides the requesting user's external ID. For each request that uses this substitution parameter, the requesting user's external ID is retrieved from the `alt_id` claim in the requesting user's access token.

For more information see ["Identity-Based Substitution" in SAS Viya Platform: CAS Authorization](#).

CAUTION

The following endpoints configuration properties should be implemented with caution as they can impact functionality:

endpoints.secured.groups

Limits non-administrators' Read access for groups. The default value is `false`.

endpoints.secured.members

Limits non-administrators' Read access for group members. The default value is `false`.

endpoints.secured.memberships

Limits non-administrators' Read access for group or user memberships. The default value is `false`.

`endpoints.secured.users`

Limits non-administrators' Read access for other users. The default value is `false`.

`sas.identities.providers.ldap`

The set of properties that are used to configure your LDAP provider.

IMPORTANT The set of properties for `sas.identities.providers.ldap` are global values that apply to all LDAP configurations.

`followReferrals` – OFF by default

Enables referrals to other servers or domains to be followed automatically.

`membershipCacheRefreshInterval`

Specifies the interval that is used to refresh the membership cache for the LDAP provider. The default value is `6h`.

`pagedResults`

Enables the LDAP server to use pagination when processing requests. Pagination is enabled by default.

`pageSize`

The number of identity requests per page to be processed by the LDAP server (if pagination is enabled). The default value is `500`.

`primaryGroupMembershipsEnabled` – OFF by default

Enables processing of primary group memberships (used only for `POSIXGroup` schema).

`sas.identities.providers.ldap.connection`

The set of properties that are used to configure your LDAP provider.

`anonymousBind`

Specifies whether Read-Only operations are performed using an anonymous (unauthenticated) context.

`customEnvironmentProperties`

The set of user-added, advanced properties for configuring the LDAP client environment.

Note: Do not enter a property name-value pair unless you are directed to do so by SAS Technical Support.

`host`

The host name of the LDAP server to connect to.

password

The password for logging on to the LDAP server. If **anonymousBind** is enabled, specify a value of **none**.

pool.enabled

Enables pooling of LDAP connections. Pooling is enabled by default.

pool.evictionTimePeriodMillis

The number of milliseconds that the idle-object evictor thread sleeps between runs. If the value is non-positive, the idle object evictor thread does not run. The default value is **240000**.

pool.idleTimeMillis

The minimum amount of time in milliseconds that objects can sit idle in the pool before becoming eligible for eviction by the idle-object evictor, if present. The default value is **480000**.

pool.maxActive

The maximum number of active connections of a given type (either Read-Only or Read-Write) that can be allocated from the pool at the same time. The default value is **8**.

pool.maxIdle

The maximum number of active connections of a given type (either Read-Only or Read-Write) that can remain idle in the pool without extra connections being released. For no limit, specify a non-positive value. The default value is **8**.

pool.maxSize

The maximum number of active connections of all types that can be allocated from the pool at the same time. For no limit, specify a non-positive value. The default value is **-1**.

pool.maxWait

The maximum amount of time in milliseconds that the pool waits for a connection to be returned before throwing an exception. For no limit, specify a non-positive value.

pool.minIdle

The minimum number of active connections of a given type (either Read-Only or Read-Write) that can remain idle in the pool without extra connections being created. To create no extra connections, specify zero. The default value is **0**.

pool.testOnBorrow

Enables validation of objects before they are borrowed from the pool. If an object fails validation, it is dropped from the pool and an attempt is made to borrow another object. This option is enabled by default.

pool.testOnReturn

Enables validation of objects before they are returned to the pool. This option is disabled by default.

pool.testWhileIdle

Enables validation of objects by the idle-object evictor, if present. If an object fails validation, it is dropped from the pool. This option is enabled by default.

pool.whenExhaustedAction

An integer that indicates the behavior when the pool is exhausted. Valid values are: **0** (fail), **1** (block), or **2** (grow).

port

The port for connecting to LDAP.

Note: When connecting via LDAP, port values are set to 389. When connecting via Lightweight Directory Access Protocol over TLS (LDAPS), port values are set to 636.

startTLS.mode

When using StartTLS, determine which mode to enable. The possible values are `none` (default) or `simple`. This property is case-sensitive.

url

The URL for connecting to LDAP.

The default is: `url: ldap://{sas.identities.providers.ldap.connection.host}:{sas.identities.providers.ldap.connection.port}`

When the host and port properties have been specified, the `url` must be changed if you are connecting via the LDAPS protocol.

userDN

The distinguished name (DN) of the user account for logging on to the LDAP server (for example, `cn=AdminUser, dn=example, dn=com`). If **anonymousBind** is enabled, specify a value of **none**.

sas.identities.providers.ldap.group (Field Mappings)

The set of properties that are used to configure the mapping of group-level fields in your LDAP provider to group-level fields in SAS. For each of the following SAS fields, you specify the corresponding field in your LDAP provider. The default values are valid for most implementations of Microsoft Active Directory. For other LDAP providers, you must provide different values for some fields.

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
<code>accountId</code>	The field in the LDAP provider that is used to populate the group's ID.	<code>sAMAccountName</code>
<code>createdDate</code>	The field in the LDAP provider that is used to populate the group's account created date.	<code>whenCreated</code>
<code>description</code>	The field in the LDAP provider that is used to populate the group's description.	<code>description</code>
<code>member</code>	The field in the LDAP provider that is used to populate the members of the group.	<code>member</code>
<code>memberOf</code>	The field in the LDAP provider that is used to populate the groups that this group is a member of. Set <code>memberOf</code> to none if the	<code>memberOf</code>

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
	LDAP group attribute is not a fully qualified DN value.	
modifiedDate	The field in the LDAP provider that is used to populate the date on which the group's account was last modified.	whenChanged
name	The field in the LDAP provider that is used to populate the group's name.	displayName
objectClass	The object class value to use when loading groups.	group

sas.identities.providers.ldap.group (Additional Properties)

The set of properties that are used to configure information for retrieving group information from your LDAP provider.

Note: If the `sas.identities.providers.ldap` property `followReferrals` is set to `OFF`, the Identities service does not process referrals.

baseDN

The point from which the LDAP server searches for groups (for example, `ou=groups,dc=example,dc=com`).

distinguishedName

The field in the LDAP provider that is used to populate the group's distinguished name value.

Note: If your LDAP server does not support an explicit distinguished name attribute (for example, OpenLDAP), you must set this property to `none`.

objectFilter

The filter for customizing results that are returned when groups are queried (for example, `objectClass=group`).

You can create a custom filter to exclude identities whose accounts are disabled or expired, or to exclude objects that represent computer resources rather than actual groups. If you have a large number of groups, using a custom filter can improve performance and reduce memory requirements. In addition, user management tasks can be performed more efficiently if only relevant identities are listed in SAS Environment Manager.

searchFilter

The filter that is used to find a group account. The default filter is
`${sas.identities.providers.ldap.group.accountId}={0}`.

sas.identities.providers.ldap.user (Field Mappings)

The following properties specify the mapping of user-level fields in your LDAP provider to user-level fields in SAS. For each of the following SAS fields, you specify the corresponding field in your LDAP provider. The default values are valid for most implementations of Microsoft Active Directory. For other LDAP providers, you must provide different values for some fields.

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
accountId	The field in the LDAP provider that is used to populate the user's ID. Note: The accountId is case-sensitive and must exactly match the user name that is provided by the authentication protocol. In addition, for each supported protocol, a different format is assumed to be the user name. For more information see "About User Names" in SAS Viya Platform: Authentication .	sAMAccountName
address.country	The field in the LDAP provider that is used to populate the user's country.	co
address.locality	The field in the LDAP provider that is used to populate the user's city.	l
address.postalCode	The field in the LDAP provider that is used to populate the user's postal code.	postalCode
address.region	The field in the LDAP provider that is used to populate the user's region or state.	region
address.street	The field in the LDAP provider that is used to populate the user's street address.	street
createdDate	The field in the LDAP provider that is used to populate the user's account created date.	whenCreated
description	The field in the LDAP provider that is used to populate the user's description.	description

Property	Description	Default (valid for most implementations of Microsoft Active Directory)
emailAddress.other	The field in the LDAP provider that is used to populate the user's alternate email address.	otherMailbox
emailAddress.work	The field in the LDAP provider that is used to populate the user's work email address.	mail
emailAddress.sms	The field in the LDAP provider that is used to populate the user's SMS email address.	None
memberOf	The field in the LDAP provider that is used to populate the groups that this user is a member of.	memberOf
modifiedDate	The field in the LDAP provider that is used to populate the date on which the user's account was last modified.	whenChanged
name	The field in the LDAP provider that is used to populate the user's name.	displayName
objectClass	The type of user objects that are being searched for.	organizationalPerson
phone.business	The field in the LDAP provider that is used to populate the user's work phone number.	telephoneNumber
phone.businessFax	The field in the LDAP provider that is used to populate the user's work fax number.	facsimileTelephoneNumber
phone.home	The field in the LDAP provider that is used to populate the user's home phone number.	homePhone
phone.mobile	The field in the LDAP provider that is used to populate the user's mobile phone number.	mobile
phone.pager	The field in the LDAP provider that is used to populate the user's pager number.	pager
title	The field in the LDAP provider that is used to populate the user's title.	title

sas.identities.providers.ldap.user (Other Properties)

The set of properties that are used to configure additional information for retrieving user information from your LDAP provider.

Note: If the `sas.identities.providers.ldap` property `followReferrals` is set to `OFF`, the Identities service does not process referrals.

baseDN

The point from which the LDAP server searches for users.

distinguishedName

The field in the LDAP provider that is used to populate the user's distinguished name value.

Note: If your LDAP server does not support an explicit distinguished name attribute (for example, OpenLDAP), you must set this property to `none`.

objectFilter

The filter for customizing results that are returned when querying users.

You can create a custom filter to exclude identities whose accounts are disabled or expired, or to exclude objects that represent computer resources rather than actual users. If you have a large number of users, using a custom filter can improve performance and reduce memory requirements. In addition, user management tasks can be performed more efficiently if only relevant identities are listed in SAS Environment Manager.

Here is an example of a filter that excludes identities that represent computers and identities that are inactive. This filter is compatible with Microsoft Active Directory.

```
( & (objectCategory=person) (objectClass=user) (!
  (userAccountControl:1.2.840.113556.1.4.803:=2) ) )
```

For OpenLDAP, the filter `(objectclass=person)` excludes identities that represent resources other than users.

searchFilter

The filter used for locating a user account in the LDAP provider so that the user can make a connection using an ID and password.

The default filter is `${sas.identities.providers.ldap.user.accountId}={0}`.

Identity Management: Troubleshooting

General Troubleshooting Strategies

For general information about how to approach issues, see [“Troubleshooting Strategies” in Troubleshooting Strategies](#).

Cannot Access Cloud Analytic Services

- If the user cannot start a CAS session, ensure that the user’s account meets all applicable requirements. See [SAS Viya Platform: Authentication](#).
- If an error message in the CAS log states that the user “failed mid-tier authentication,” the user’s credentials are not valid for your direct LDAP provider. See the discussion of dual authentication in [SAS Viya Platform: Authentication](#).
- Ensure that users have a host account before adding them to the CASHostAccountRequired group. A member of the CASHostAccountRequired group without a host account cannot start the necessary CAS session.

If a user is in SCIM mode, they need to define CAS_CLOUDNATIVE=1 (currently the default value) to ensure that cas is in SCIM mode as well.

Cannot View Users and Group Members

If LDAP is enabled, and you receive an error while viewing users, groups, or their memberships from SAS Environment Manager or any other client, then an LDAP referral might have been encountered. The SAS Viya platform does not process LDAP referrals.

Here is an example of this error message:

```
Load Users
An error occurred loading the list of users.
exception:
org.springframework.ldap.PartialResultException
Caused by: javax.naming.PartialResultException: Unprocessed Continuation
Reference(s); remaining name 'DC=COMPANY,DC=COM'
```

This occurs because LDAP is initialized based on what the Identities service itself configures only. Therefore, any environment variables that are set are not processed. Connecting to the global catalog might be a viable solution.

Cannot Access Esri Geographic Mapping Resources

Ensure that the user is a member of the Esri Users group. Users that are members of the Esri Users group have access to tokens for which there is a fee. See [“Predefined Custom Groups” on page 47](#).

Cannot Retrieve List of Users or Groups

If the following error occurs while attempting to retrieve a list of users or groups that are defined in your environment, then this is due to a failed LDAP search by the Identities service:

```
[LDAP: error code 12- Unavailable Critical Extension]
```

The Identities service attempted an LDAP search for a collection of users or groups, but the request failed because the LDAP server does not support paged queries.

To resolve this issue, follow these steps to change the value of the `sas.identities.providers.ldap.pagedResults` configuration property:

- 1 Log on to SAS Environment Manager as an administrator.
- 2 Navigate to the **Configuration** page. From the **View** drop-down list, select **Definitions**. In the **Filter** field, enter `sas.identities.providers.ldap`.
- 3 From the **Identities service** drop-down list on the right pane, click . Change the `pagedResults` property to `off`.
- 4 Click **Save**.

Cannot Update User Information

Any service or application that uses the Identities service pulls the associated user information from the LDAP server directly. Therefore, user information such as phone number, work address, and email address cannot be updated in the SAS Viya platform but must be updated in LDAP.

For example, to specify a different email address to receive SAS Visual Analytics alerts, the email address field must be updated directly in LDAP.

Cannot Log In to the SAS Viya Platform

Ensure that no two users have the same email address in LDAP. Users might have problems logging in to the SAS Viya platform if another user has the same email address.

Membership in a sasapp LDAP, Custom, or Host Group Is Ignored

The SAS Viya platform reserves the group identifier **sasapp** for internal use by services. Only services are members of the privileged internal group **sasapp**. If you also have a **sasapp** host group, LDAP group, or custom group, unintended results can occur. Descriptive information (such as the Authorization window and the Users page in SAS Environment Manager) reflects membership in the group. However, actual access does not reflect membership in the group. Here are details:

- When a user who is a member of a **sasapp** LDAP or custom group signs in to the SAS Viya platform, SAS Logon Manager discards the user's **sasapp** membership information, excluding it from the user's OAuth token. By discarding that membership information, SAS Logon Manager ensures that the privileges of the **sasapp** internal group are not made available to users.
- If you specify the **sasapp** group as the principal in a general authorization rule, that rule affects only the **sasapp** internal group.
- When a user who is a member of a **sasapp** host group authenticates to CAS, CAS alters its copy of the user's membership information, replacing the **sasapp** group name with the group ID. By altering that membership information, CAS ensures that the privileges of the internal group **sasapp** are not made available to users.
- If you specify the **sasapp** group as the principal in a CAS access control, that access control affects only the **sasapp** internal group.

Note: The SAS Viya platform does not currently prevent the creation of a **sasapp** custom group. The SAS Viya platform cannot prevent the creation of a **sasapp** host or LDAP group.

Resolving a 409 Conflict When Querying a User or Group Identifier

It is possible to get a 409 conflict exception on the Identities Service identifier endpoints when multiple users or multiple groups have a generated identifier collision. A 409 conflict exception can occur when user or group generated UID or GID values conflict with the stored values for another user or group in the Identities service database. The endpoints that can return this 409 conflict are listed below:

- identities/groups/{groupID}/identifier
- identities/users/{userID}/identifier

This issue is corrected by manually overriding the value on the identity getting the 409 exception. You should define environment variables that allow the administrator to override the UID or GID generation value if a collision between IDs occurs. The environment variables that provide support for the override process are listed below.

- (For group conflicts) `sas.identities.identifier.groups.override`
- (For user conflicts) `sas.identities.identifier.users.override`

IMPORTANT It is important to note that this override process is allowed or used only if you have configured the environment to be non-generating. You cannot override conflict values if your environment is in generate mode:

When the properties are defined: The Identities service attempts to set the values for the specified user to the corresponding specified override values. Namely, if the endpoint called is `identities/users/{userID}/identifier`, then the override property is checked. If it exists, the Identities service determines whether the ID in the override string matches the user ID on the request. If so, then the Identities service overrides the UID or GID value with the value specified for that user ID in the `sas.identities.identifier.users.override` property string.

Example properties:

- `sas.identities.identifier.users.override: testuser=1500002`
- `sas.identities.identifier.groups.override: testgroup=1600002`

The properties need to be set up so that they can be parsed properly.

The format of the variable is `id1=value1,id2=value2,...`

The comma (,) is the default delimiter value used between IDs to override.

The default delimiter can be changed if required. (See below for additional details).

The UID/GID value must be numeric.

A further example using multiple overrides where the delimiter (,) comes into play is as follows:

- `sas.identities.identifier.users.override:`
`testuser=1500002,testuser99=2000000,testuser200=3000000`

To change the delimiter used from a comma (,) when specifying multiple overrides, there is an additional property that can be defined:

- `sas.identities.identifier.override.delimiter`

Example delimiter override: `sas.identities.identifier.override.delimiter: "&"`

This would require the override variables to now be in the form: `id1=value1=value2,...`

Example: `sas.identities.identifier.users.override:`

`testuser=1500002==2000000=3000000.`

Cannot Perform Sync Due to Error

When you see this error message, it is because there is no SAS token or oauth token present and the resource is not public.

Identity Management: Interfaces

In the following table, the shaded part of each circle is an approximation of the amount of user management functionality that a particular interface exposes. The shading indicates relative coverage. The shading does not indicate alignment of functional coverage across interfaces.

 SAS Environment Manager	A graphical enterprise web application. See “Users and Groups Page” in SAS Environment Manager: User’s Guide .
 Access Control action set	A programmatic interface for SAS (the CAS procedure), Python, R, and Lua. See “Access Control Action Set” in SAS Viya Platform: System Programming Guide .
 Command-line interface	A simple scriptable interface that provides commands for managing identities. See SAS Viya Platform: Using the Command-Line Interface .