Identity Manager 4.5

Driver for Google Apps Implementation Guide

March 29, 2016

NetIQ

Legal Notice

Contents

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate — day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with — for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit http://community.netiq.com.

About this Book and the Library

The *Driver for Google Apps Administration Guide* provides conceptual information about installing, configuring and customizing the Google Apps Driver for Identity Manager. This book defines terminology and includes implementation scenarios.

Intended Audience

This book provides information for individuals responsible for understanding administration concepts and implementing the Google Apps Driver for Identity Manager.

Other Information in the Library

The library provides the following information resources:

Installation Guide

Provides detailed planning and installation information.

User Guide

Provides conceptual information about DRA and ExA. This book also provides an overview of the user interfaces and step-by-step guidance for many administration tasks.

Trial Guide

Provides product trial and evaluation instructions and a product tour.

Help

Provides context-sensitive information and step-by-step guidance for common tasks, as well as definitions for each field on each window.

1 Overview

Identity Manager 4.5 offers automatic provisioning and synchronization of users to cloud applications. The new Google Apps driver for Novell Identity Manager can seamlessly provision and de-provision users, groups, organizational units, and contacts to the Google Apps cloud application keeping the user identity information consistent across the Identity Vault and the Google Apps domain. The Google Apps driver supports secure password synchronization across Identity Vault and Google Apps. The Google Apps driver for Identity Manager is a Subscriber channel only driver and offers out-of-the box random password generation policy for the newly provisioned users. The Google Apps driver uses a combination of language and protocols to enable identity provisioning and data synchronization between an Identity Vault with Google Apps Driver.

This section contains the following information:

1.1 Driver Concepts

1.1.1 Data Transfer between Systems

IDM drivers support two data transfer channels between the Identity Vault and the connected system, called the Publisher and Subscriber channels. The Publisher channel handles data and events from the connected system into the Identity Vault. The Subscriber channel handles data and events from the Identity Vault into the connected system.

The Google Apps driver only supports data transfers from the Identity Vault into Google Apps. Communication is one-way only.

The Publisher Channel

The Publisher Channel is not currently supported by this driver.

The Subscriber Channel

- Monitors the Identity Vault for new objects and changes to existing objects.
- Any relevant changes are sent to the shim to be executed in the Google Apps system.

Through the use of filters and policies, the driver can be configured to control and manage what changes are detected and sent to Google Apps.

1.1.2 How the Driver Works

The following diagram illustrates the data flow between Identity Manager and Google Apps API's:

Figure 1-1 *Google Apps Driver Data Flow*

The Identity Manager engine uses XDS, a specialized form of XML, to represent events in the Identity Vault. Identity Manager passes the XDS to the driver policy, which can consist of basic policies, DirXML Script, and XSLT style sheets.

After driver policy has been applied, the driver shim communicates securely over https to the Google Apps API's for your domain. The results are then communicated back to the driver. The driver then processes that information converting it into an appropriate XDS that is reported back to the Identity Manager engine.

1.1.3 Understanding The Google API's

Google has many different API's available for managing data into and out of the many different Google applications. The 4.1.x driver supports the following API's:

- Directory API - The Directory API is responsible for creating user, group and organization unit objects.
- Contact API* - The contact API is similar to the Profile API with the exception that it will create a Shared Contact inside of the Address Book (Contacts).
- Groups Settings API - Manage security settings, archive properites, and moderation settings of group objects.
- EMail Settings API - The email API allows modification to the default behavior (as set in your Google apps domain) for items related to email.

NOTE: * The Contact Add events do not show in the Google Apps Control Panel and Address Book (Contacts) for up to 24 hours. Modify events will show immediately.

1.2 Support for Standard Driver Features

The following sections provide information about how the Google Apps driver supports these standard driver features:

- Section 1.2.1, "Local Platforms," on page 11
- Section 1.2.2, "Remote Platforms," on page 11
- Section 1.2.3, "Supported Operations," on page 11

1.2.1 Local Platforms

A local installation is an installation of the driver on the Metadirectory server. The Google driver can be installed on the operating systems supported for the Metadirectory server.

For information about the operating systems supported for the Metadirectory server, see "System Requirements" in the *Identity Manager 4.0.2 Framework Installation Guide*.

1.2.2 Remote Platforms

The Google Apps driver can use the Remote Loader service to run on a server other than the Metadirectory server. The Google Apps driver can be installed on the operating systems supported for the Remote Loader.

For information about the supported operating systems, see "System Requirements" in the *Identity Manager 4.0.2 Framework Installation Guide*.

1.2.3 Supported Operations

The basic configuration files for the Google Apps driver are capable of performing the following operations.

- User Objects - Add, Modify, Delete, Query, Rename, set/change password, and Move
- Group Objects - Add, Modify, Delete, Query
- Organization Objects - Add, Modify, Delete, Query
- Contact Objects - Add, Modify, Delete, Query

Additional Packages add support for:

- Entitlements: User-Account and Group Membership.
- User Placement: Mirrored and Entitlement based placement..

2 Installing the Driver Files

By default, the Google Apps driver files are installed on the Identity Manager server at the same time as the Identity Manager engine. The installation program extends the Identity Vault's schema and installs the driver shim, dependent jars, and the driver packages. It does not create the driver in the Identity Vault (see Chapter 3, "Creating a New Driver," on page 15).

2.1 Installing the Driver Files

If you performed a custom installation and did not install the Google Apps driver on the Identity Manager server, you have two options:

- Install the files on the Identity Manager server, using the instructions in "System Requirements" in the *Identity Manager 4.0.2 Framework Installation Guide*.
- Install the Remote Loader (required to run the driver on a non-Identity Manager server) and the driver files on a non-Identity Manager server where you want to run the driver. (See "System Requirements" in the *Identity Manager 4.0.2 Framework Installation Guide*)

You must install the Google Apps driver on a server that has direct access to the Google Apps domain. The driver does not support running behind an HTTP Proxy server. This can be an existing Identity Manager server or a non-Identity Manager server that meets the system requirements for running the Remote Loader service (See "System Requirements" in the *Identity Manager 4.0.2 Framework Installation Guide*).

3 Creating a New Driver

After the Google Apps driver files are installed on the server where you want to run the driver (see Chapter 2, “Installing the Driver Files,” on page 13), you can create the driver in the Identity Vault. You do so by importing the driver configuration file and then modifying the driver configuration to suit your environment.

The following sections provide instructions to create the driver:

3.1 Configuring Google API Authentication

All of the Google services used by the Google Apps Driver are authorized using OAuth2 via a Service Account Flow.

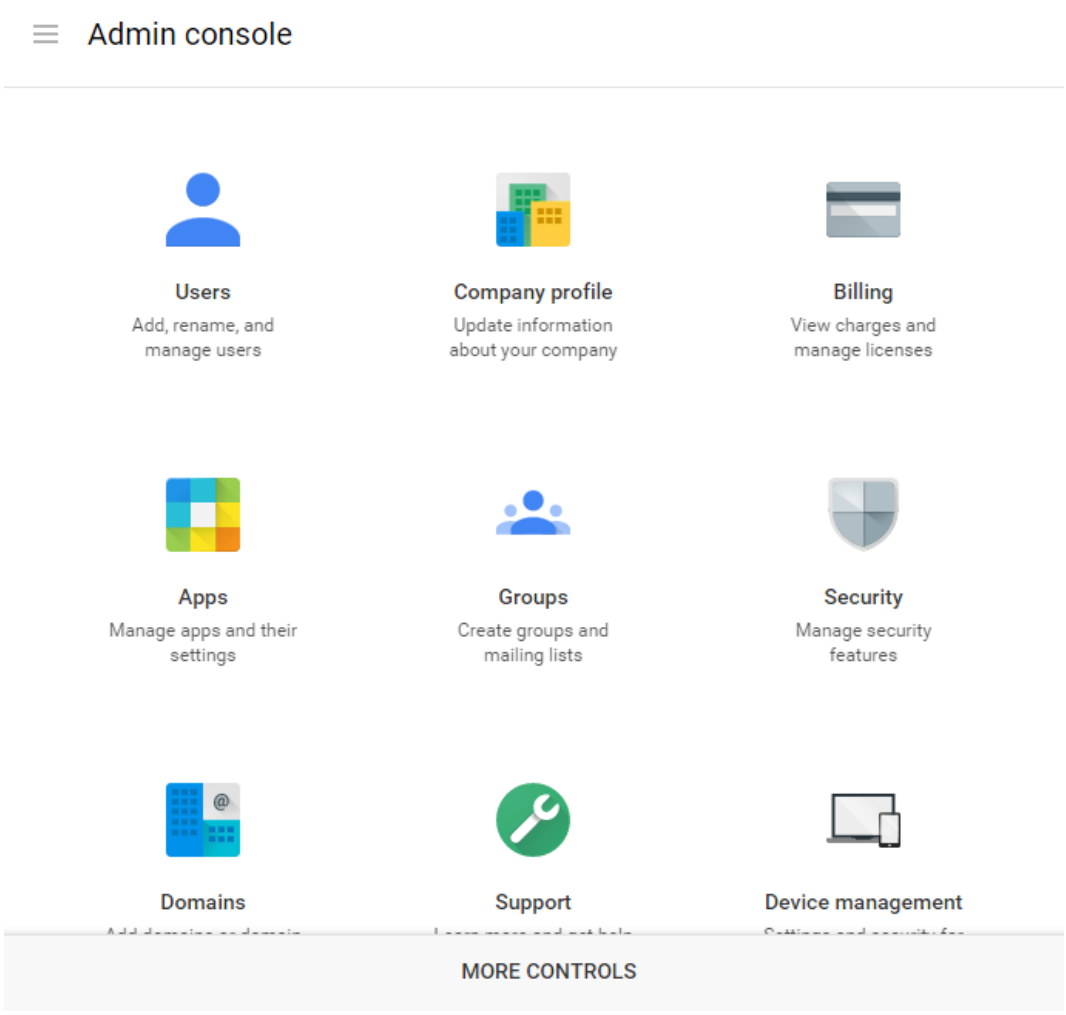
NOTE: In order to use a Service Account credential you must have an administrative user account available.

3.1.1 Creating Google Administrative Account

In order to be able to configure OAuth2 and properly authorize a Service Account Credential, a Google Apps account with Super Admin access will be required.

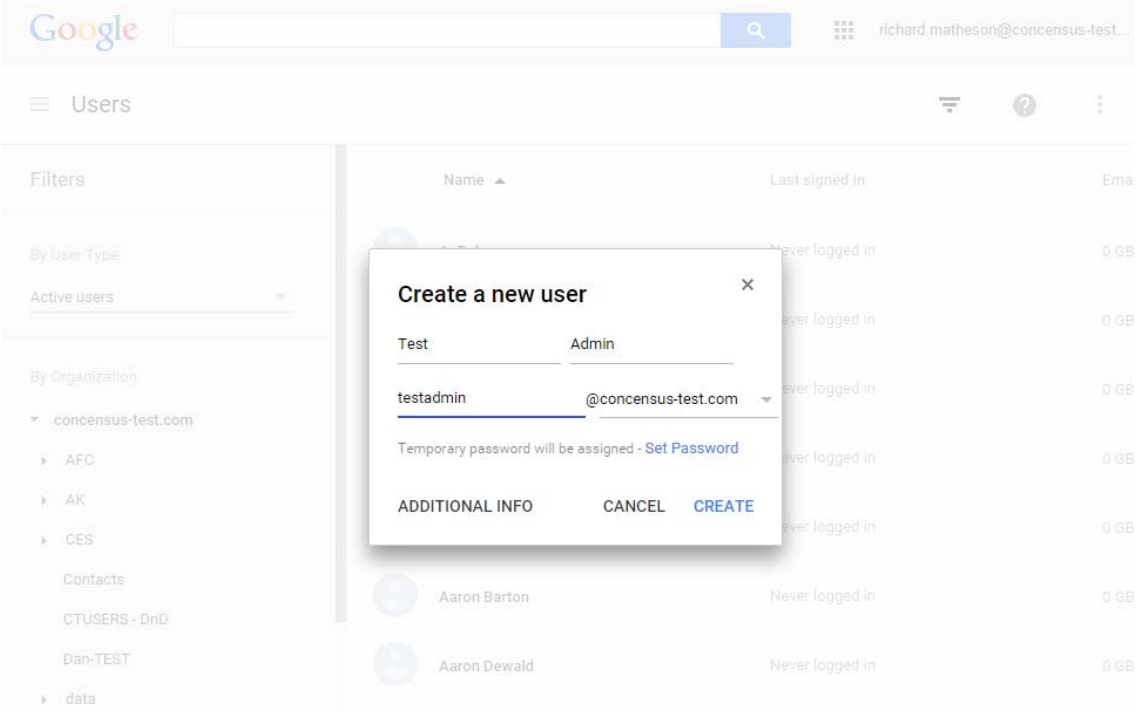
To create a new admin in the Google Domain:

1. Using a web browser, log into https://admin.google.com/AdminHome?hl=en&pli=1&fral=1



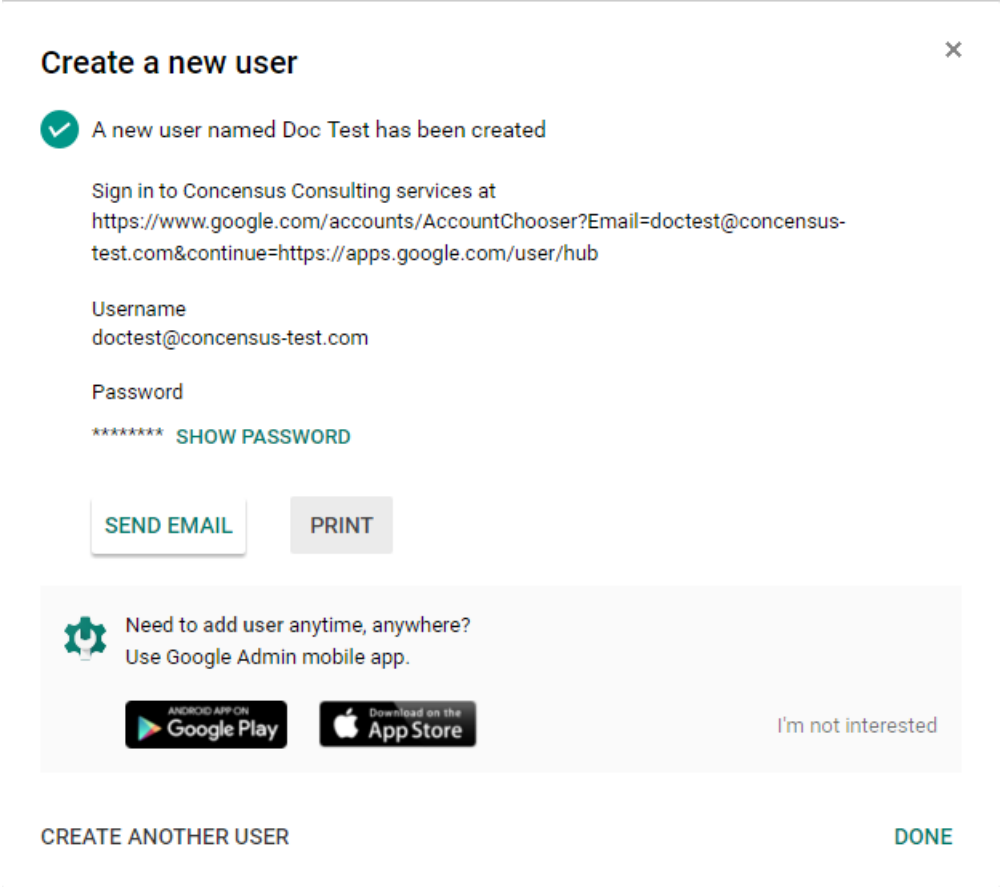
2 From the Admin Console select **Users**

3 Click on the circle in the bottom right corner with a +. Click **Add User.**



4 Enter First Name and Last Name. Set a password and additional information as desired.

5 Click **Create** to create the new user. Google will display the results.



6 Search for the new User ID in the list of Users and select it.

7 Scroll down and select **Show More**

8 Scroll down to **Admin Roles and Privileges** heading and click to expand it.

9 Click on **Manage Roles** button.

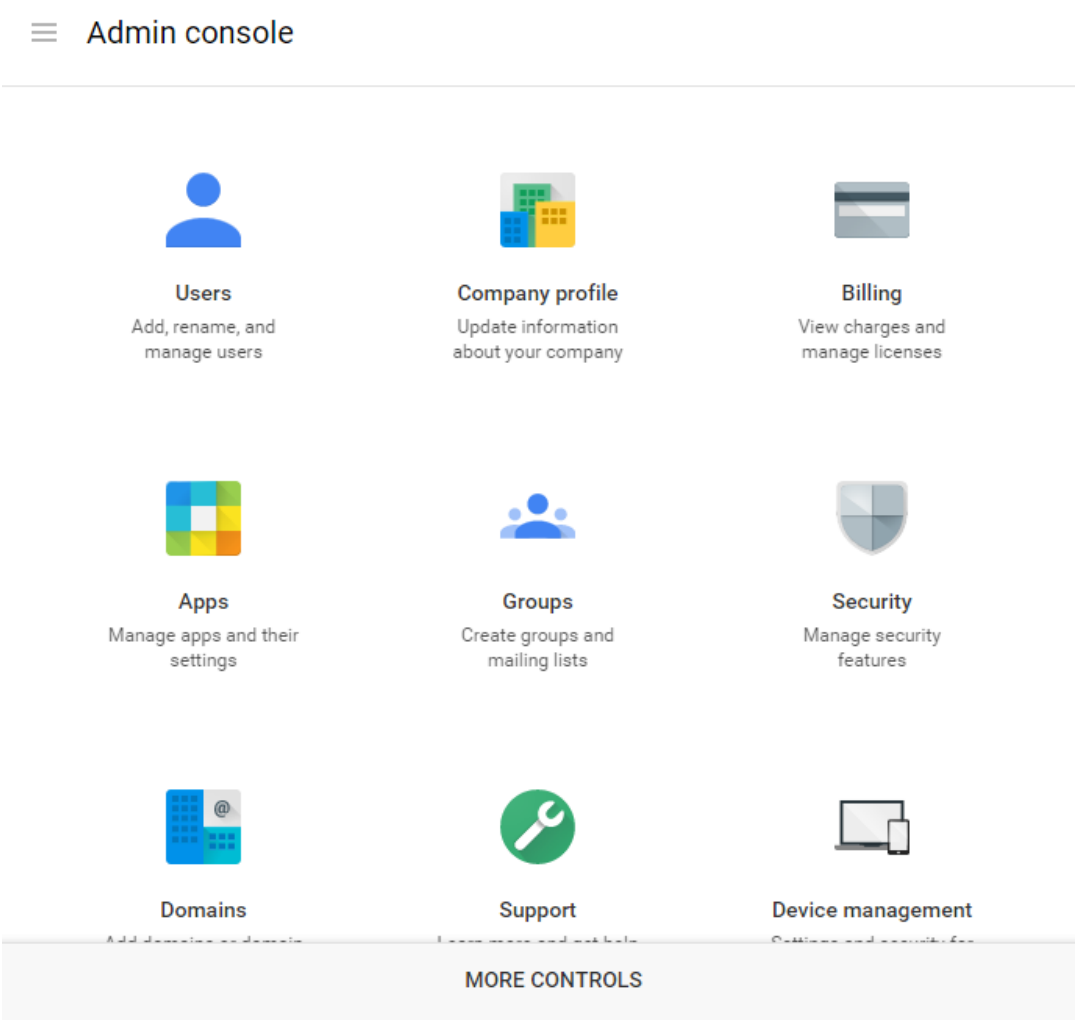
10 Click on the **Super Admin** checkbox and push **Update Roles**

11 Log out of the Google Console and log back in using the User ID you just created.

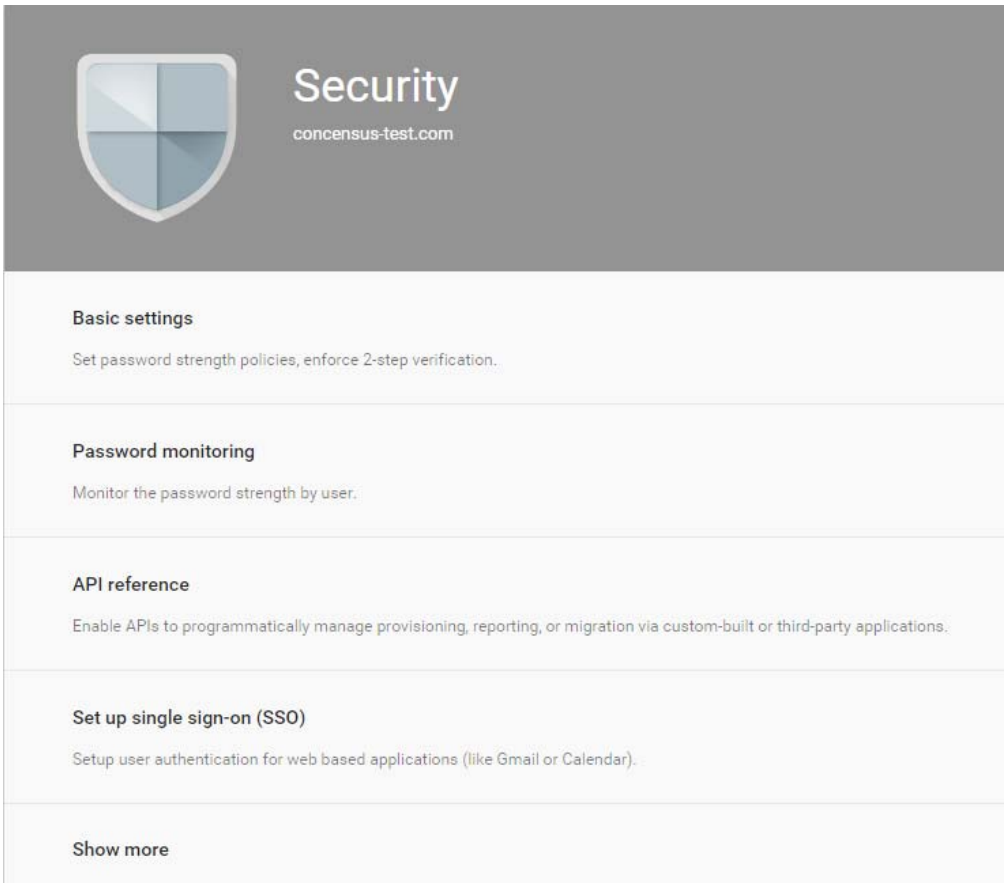
3.1.2 Enabling Google API Access

The driver will provision Users, Groups, Organizations and Shared Contacts into Google Apps. It is necessary to enable API access in your Google Apps domain before the driver can work on your domain.

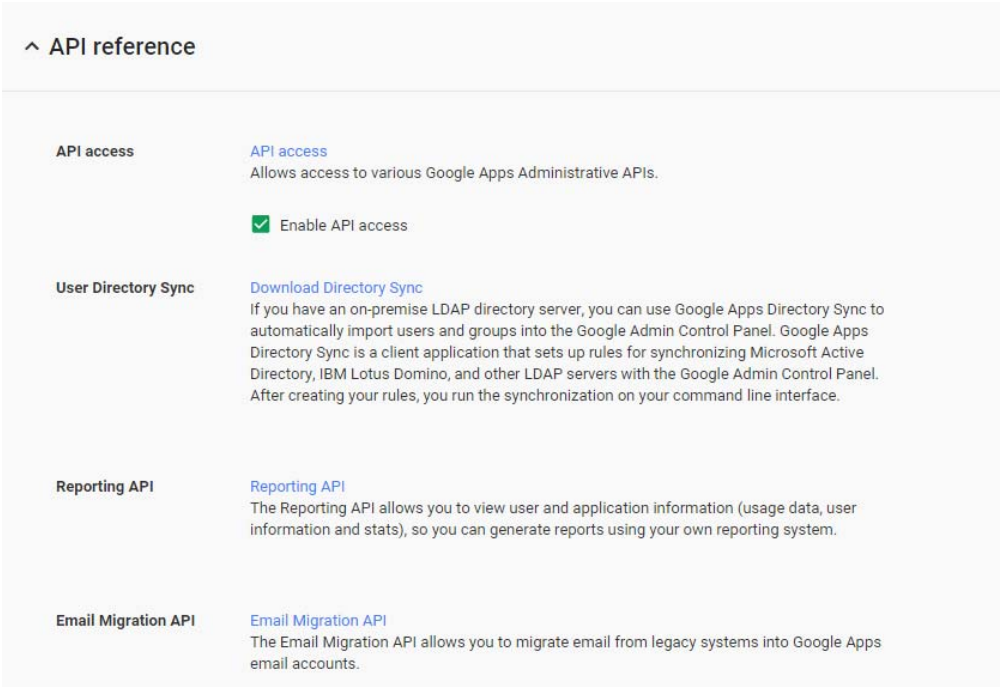
1 Using a web browser, log into the Google Apps Administration Console.



2 From the Console select **Security.**



3 From the **Security** management page, select **API Reference**



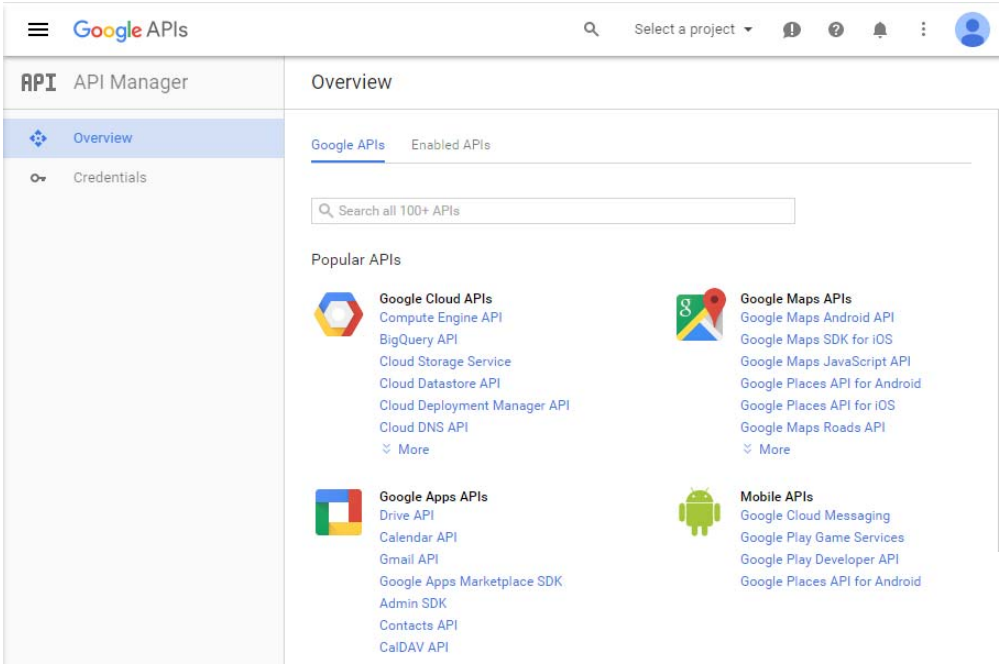
4 Check the box labeled **Enable API Access.**

3.2 Configuring OAuth2 authentication for Google APIs

NOTE: The Google Developer Console and Administrative Console change frequently as Google implements new features or rolls out updates to various services. Your view may differ from the screen shots in this section.

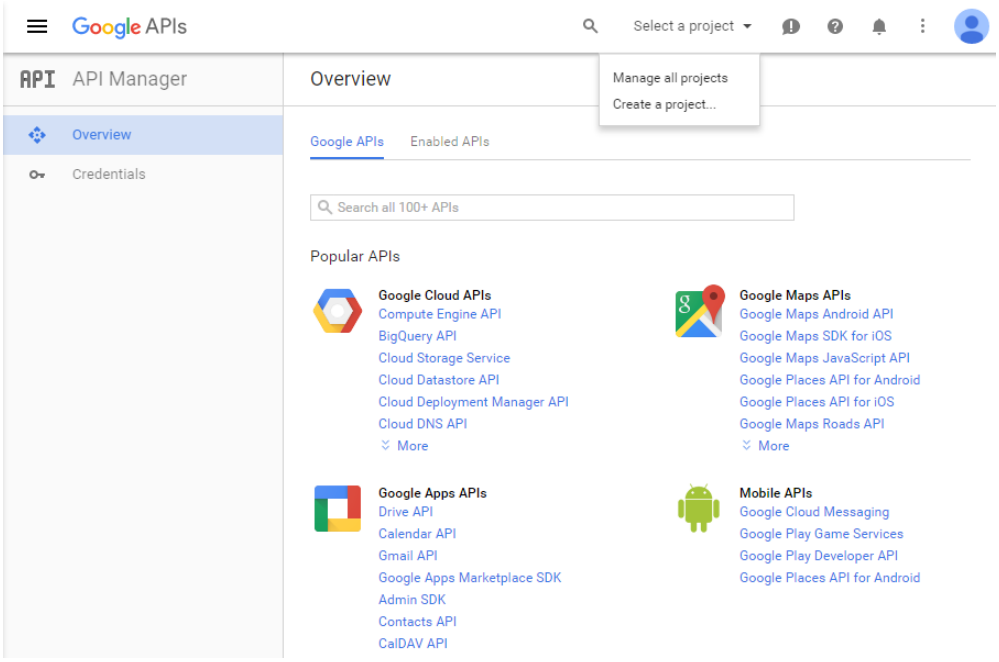
3.2.1 Creating a Google Service Account

1 Go to Google Developer Console at http://console.developers.google.com/project

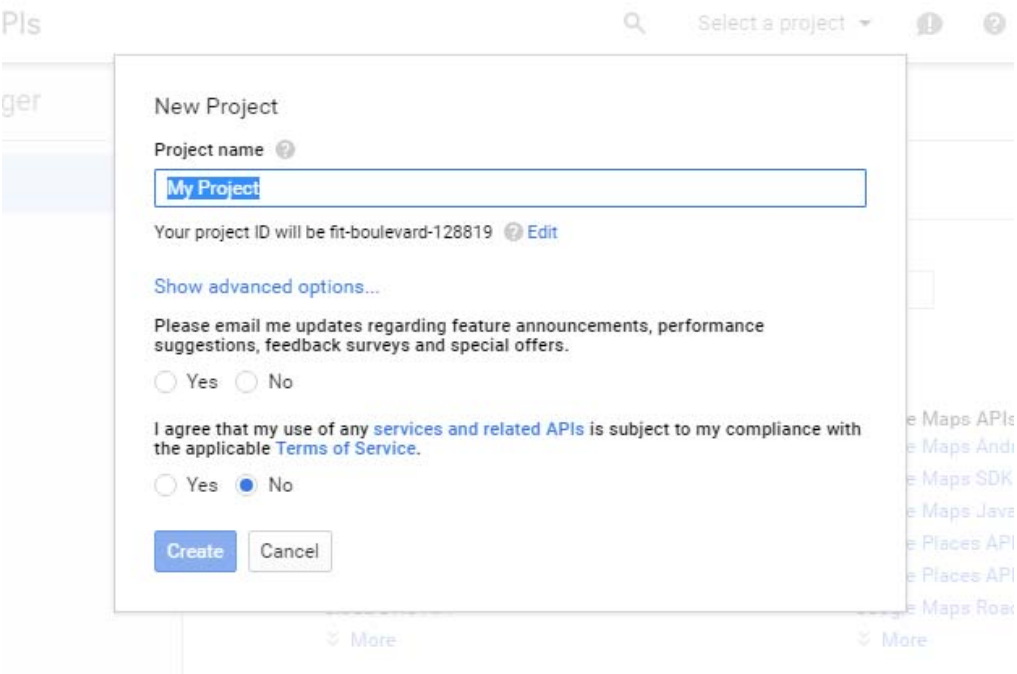


2 To create a new project, click on **Select a project** on the upper right side of the page.

NOTE: If you have already created projects, they will also be displayed in this drop-down list. You may pick an existing project to manage here.



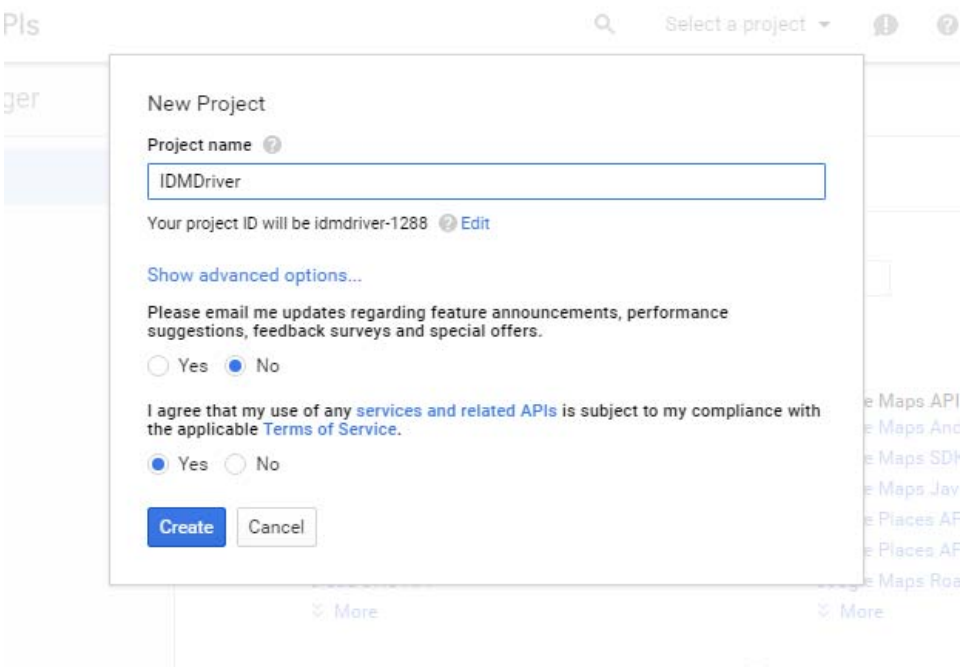
3 Click on **Create Project**



4 Fill in the **Project Name** field. The **Project ID** field will be generated by Google.

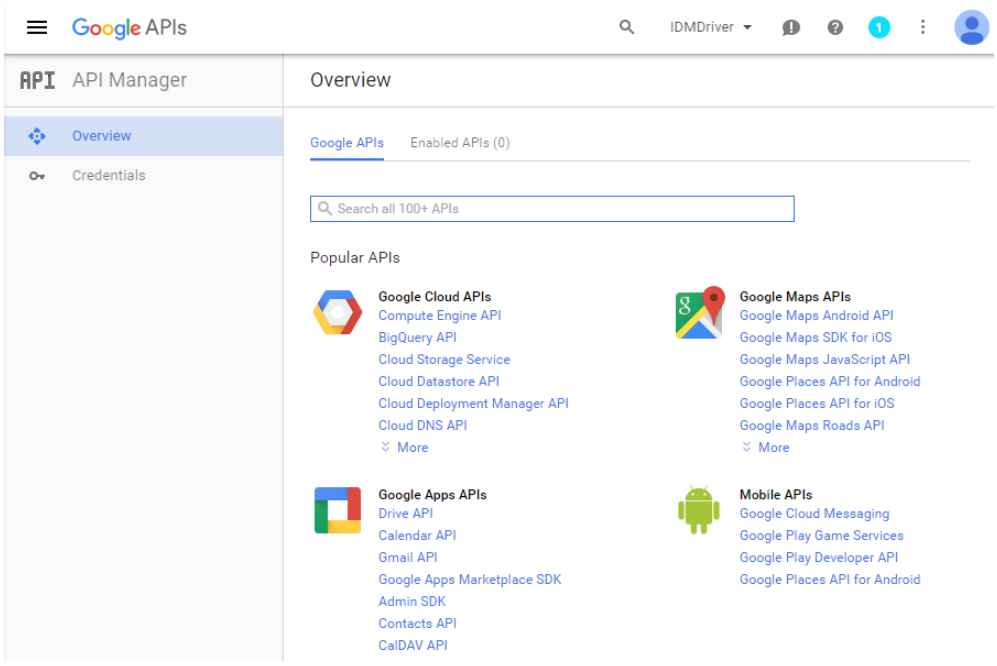
Clicking on **Show advanced options...** will allow you to select a geographic App Engine location.

NOTE: The first time a project is created on a domain Google may display additional prompts, such as opting into API email lists or accepting API terms of service.

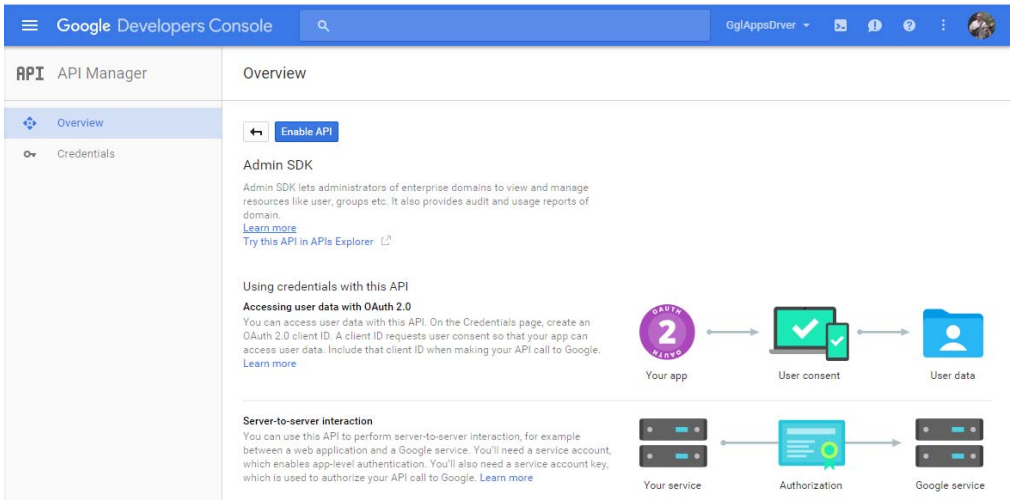


5 Click **Create.** The new project may take 1 to 2 minutes to be created.

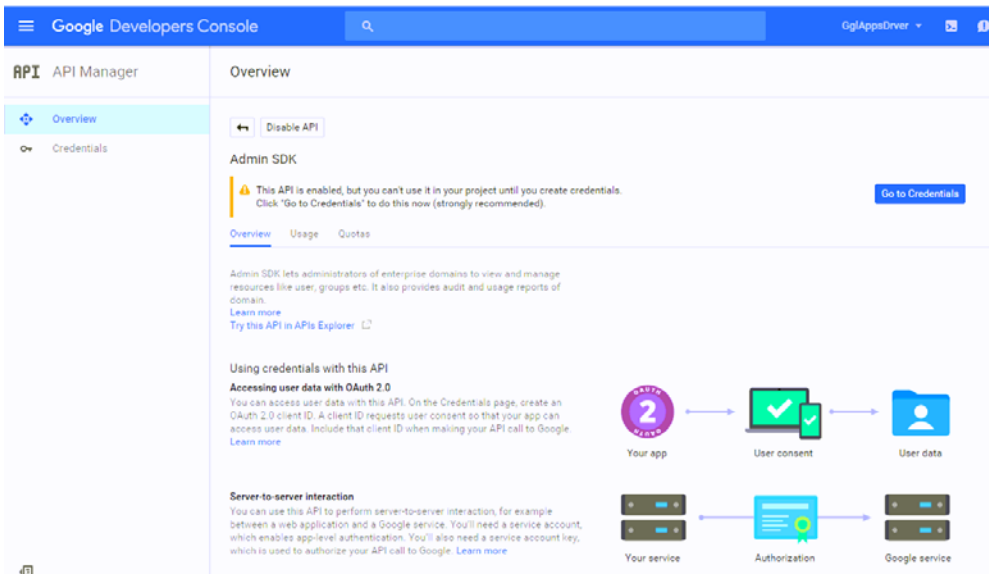
6 Once the new project has been created. The Developer Console will display options for the new project.



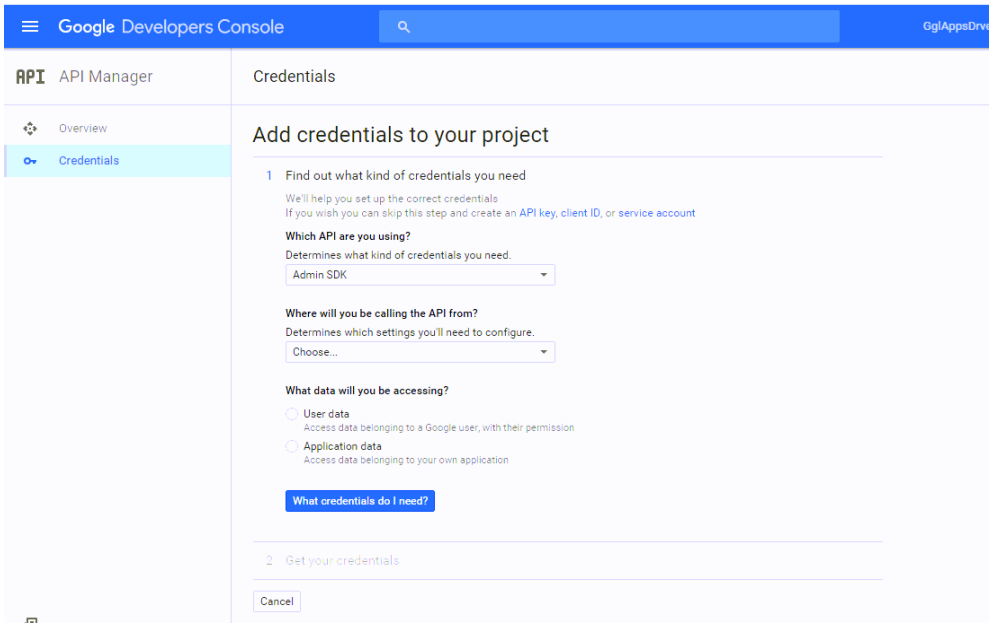
7 Click on **Admin SDK** under **Google Apps APIs**



8 Click on **Enable API**

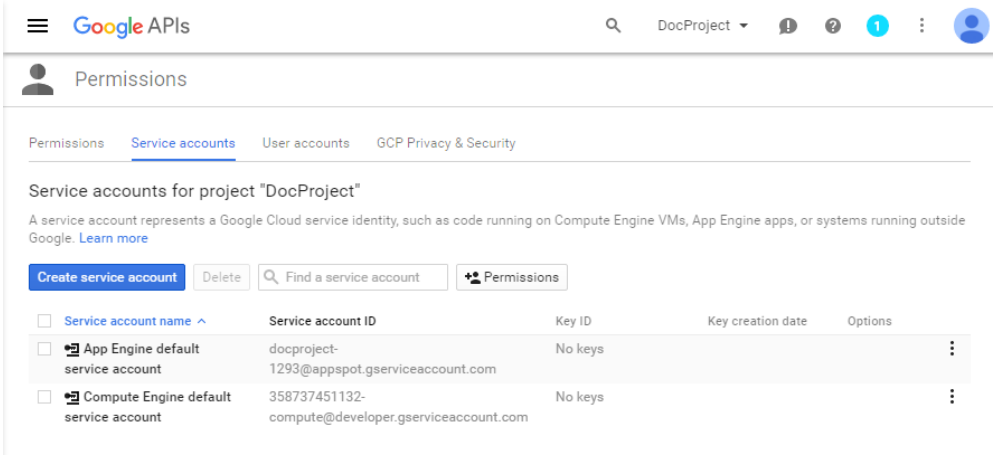


9 Click on **Go to Credentials** to create credentials now.

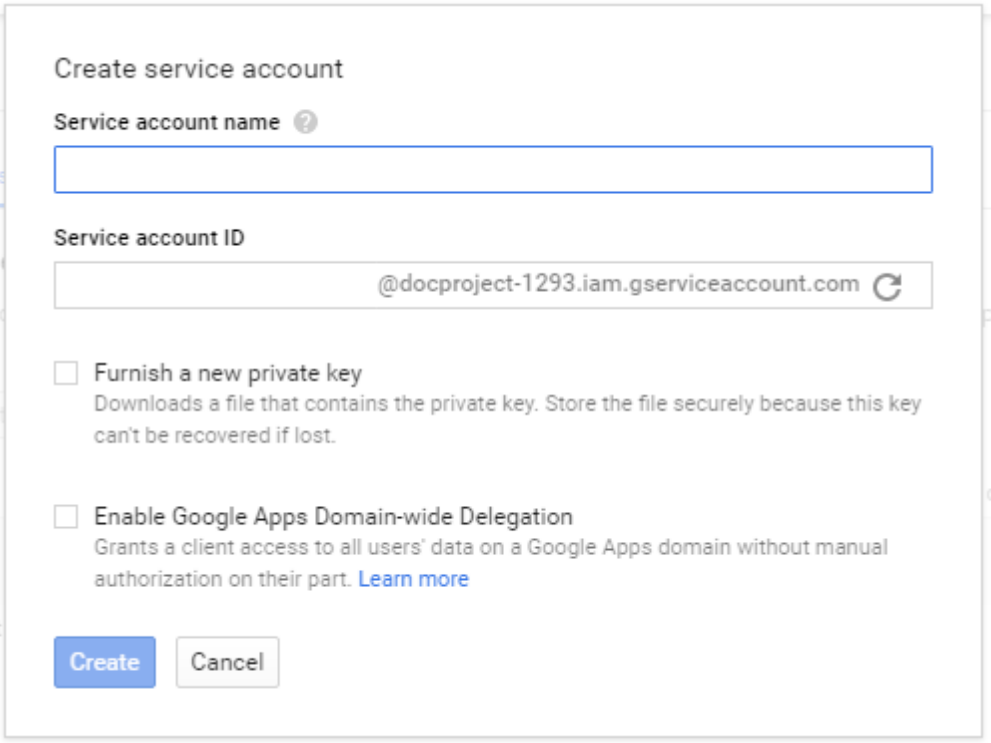


9a The Google Apps driver accesses the Google Admin SDK via a Service Account Credential. Click on the **service account** link under **Find out what kind of credentials you need.**

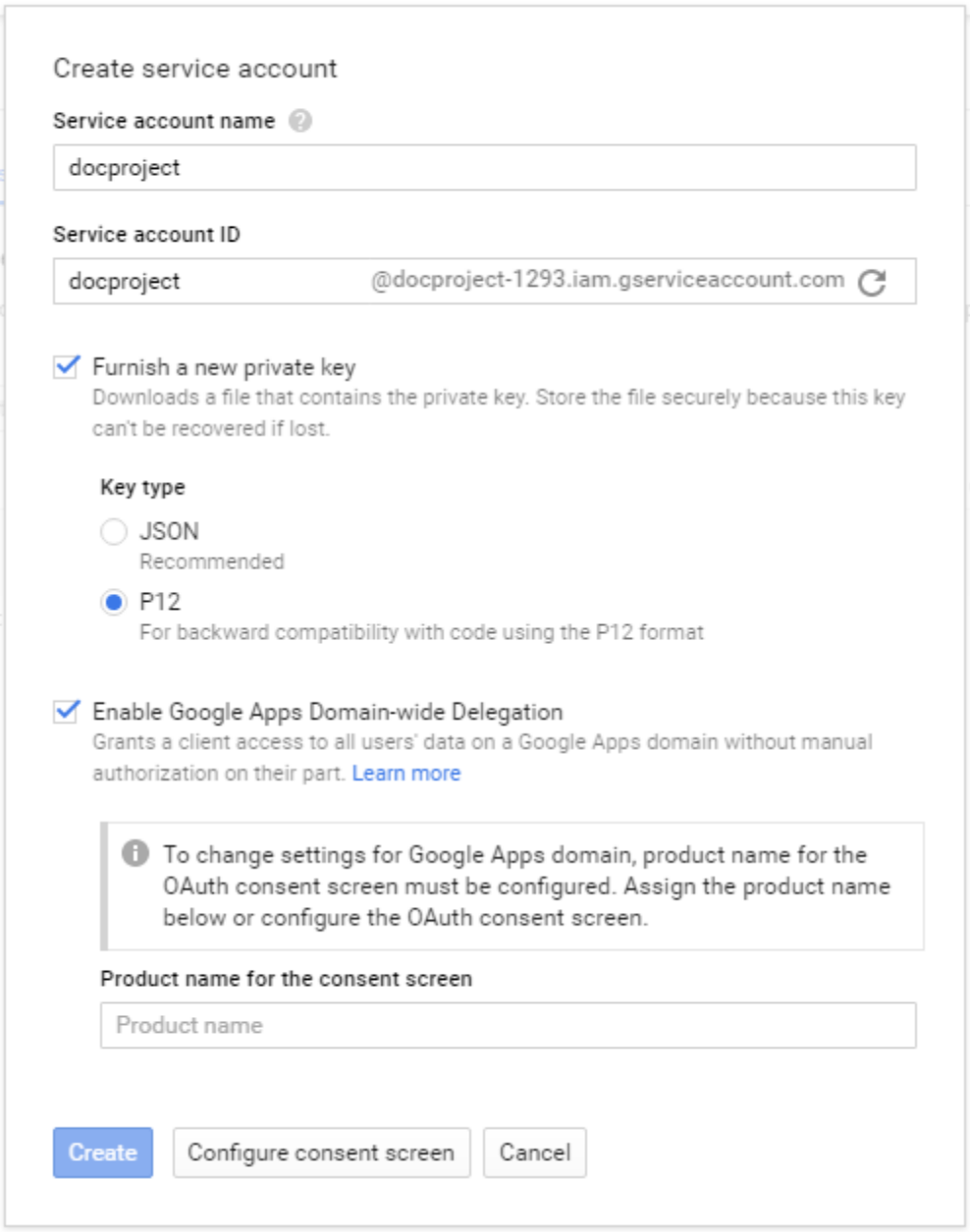
9b Click on **Create Service Account**



9c Enter the name you want to use for the service account.



NOTE: Google automatically populates the value of the Service Account ID. You will need to save the value of the Service Account ID for use in configuring the driver.

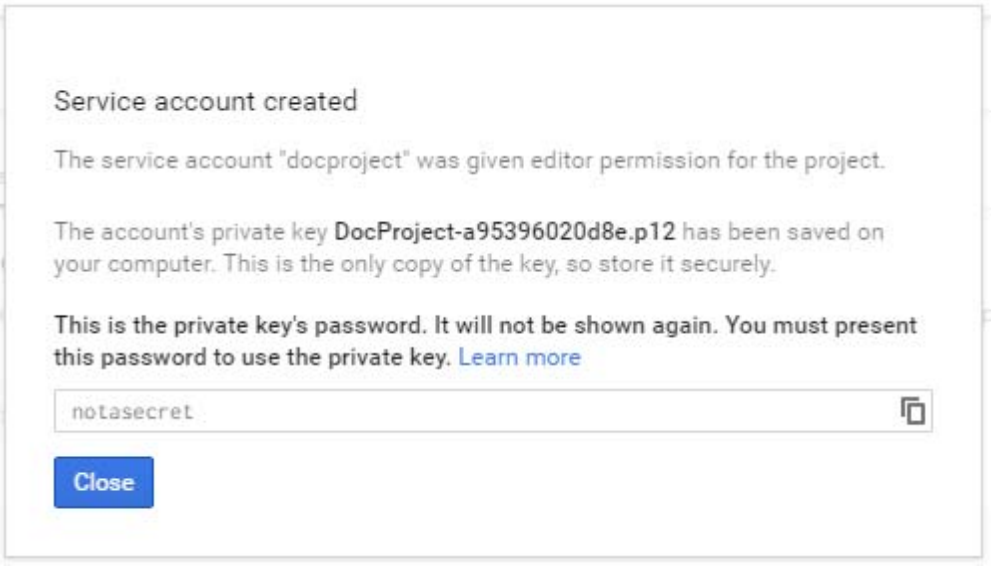


9d Check the box for **Furnish a new private key** and select **P12** as the key type.

9e Check the box for **Enable Google Apps Domain-wide Delegation.**

9f Enter a value for **Product name for the consent screen.**

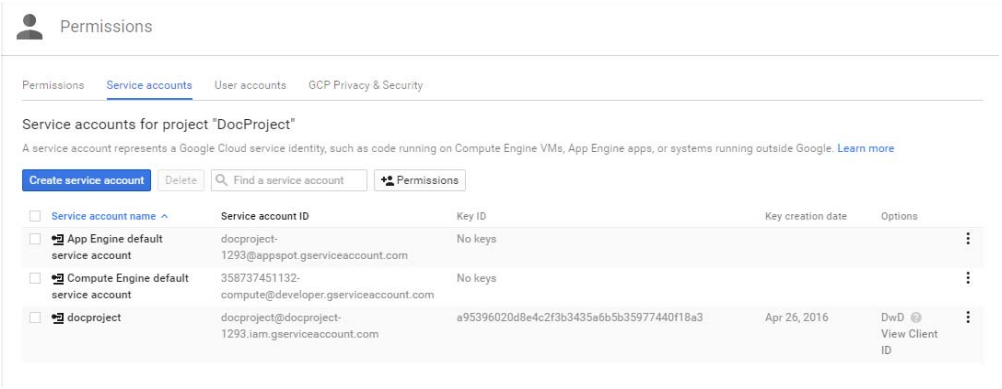
9g Click **Create**



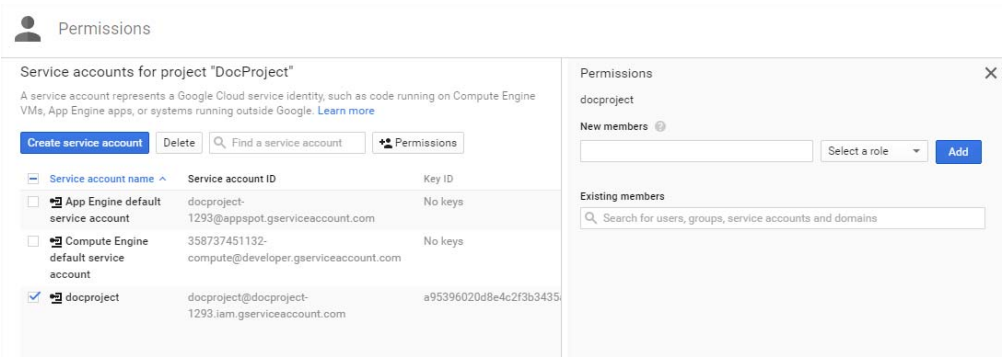
NOTE: As part of the service account creation process Google creates and downloads the P12 file for your service account to your computer. Please verify that a file with the name shown in the confirmation screen exists in your browser's download folder.

9h Press **Close**

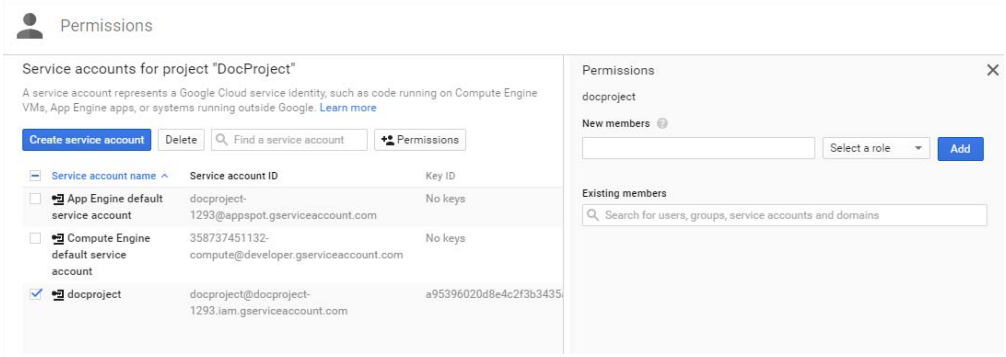
9i The service account is created and Google shows the Permissions screen for Service Accounts. You will need to have the Email address and Client ID shown on this page when configuring the driver.



9j Click on the **Permissions** button.



9k As a Best Practice, Google recommends that you create at least one additional owner for the project.

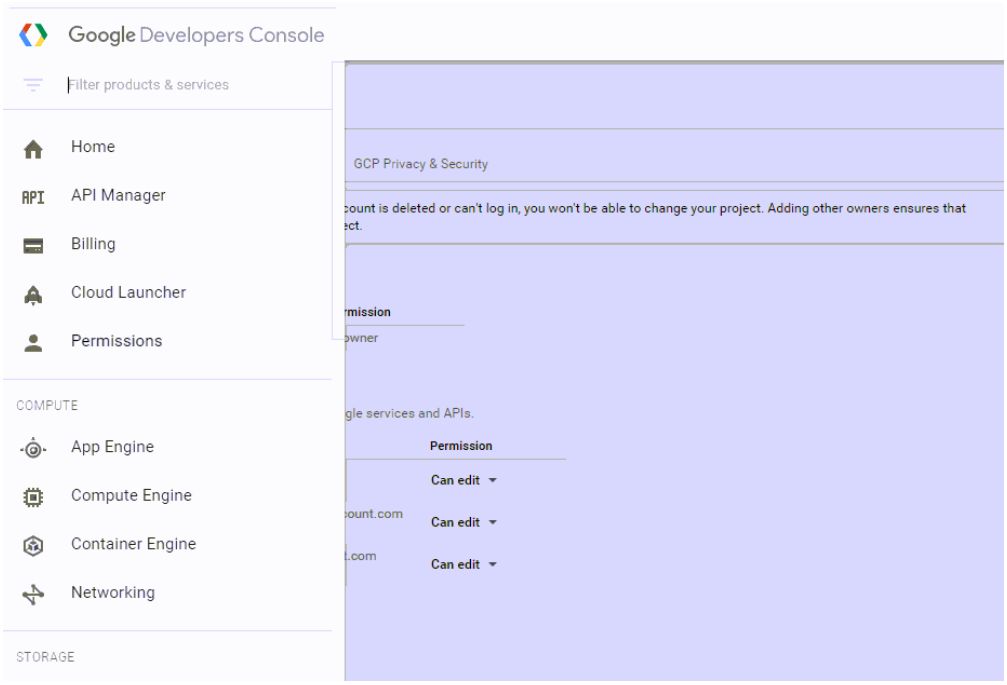


9l Enter the email address for the Google account to be added as owner in **New members**.

9m Click on the **Select a role** drop-down list and select a role. The options are **Owner, Editor, Viewer, Service Account Actor**. Select **Owner**

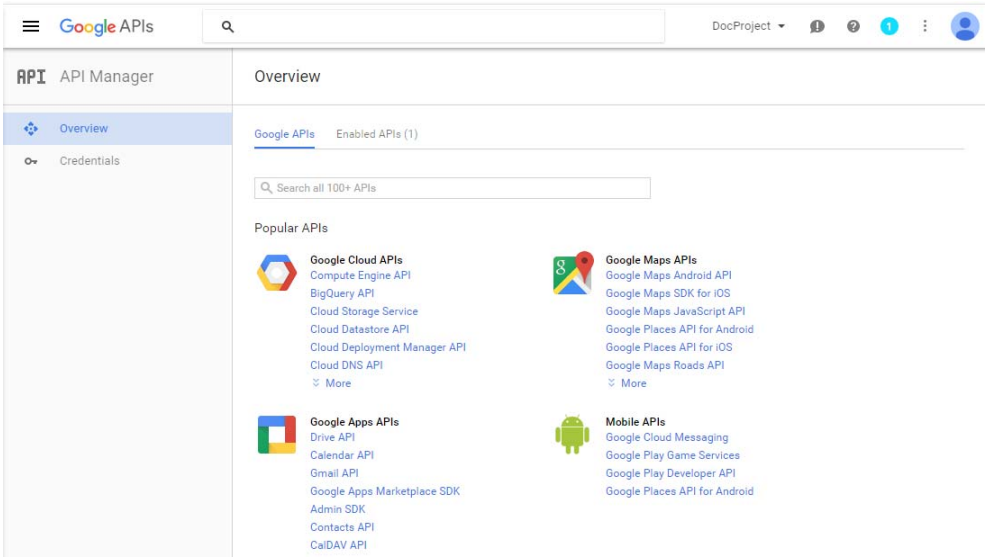
9n Click **Add**

9o Return to enabling Google APIs required by the Google Driver. To do this click on the three horizontal lines to the left of **Google Developer Console**

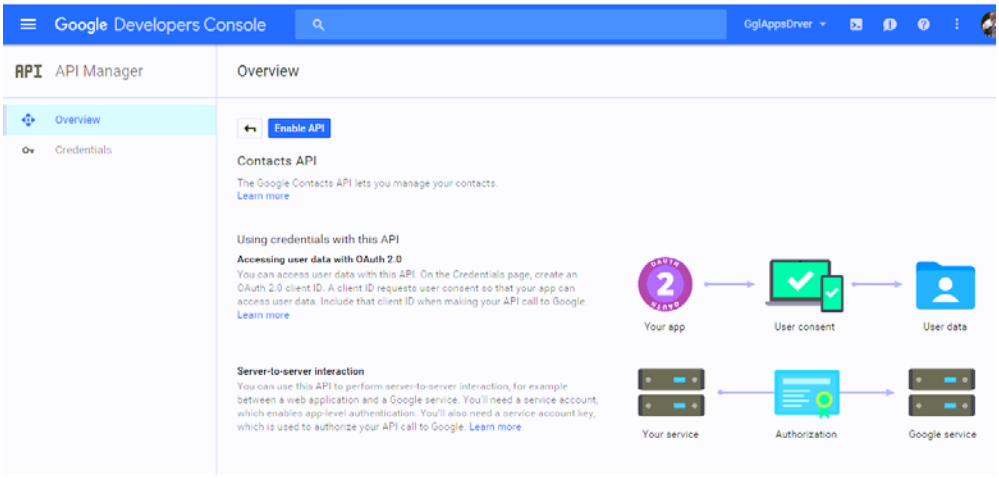


9p Select **API Manager**

10 Continue enabling Google APIs



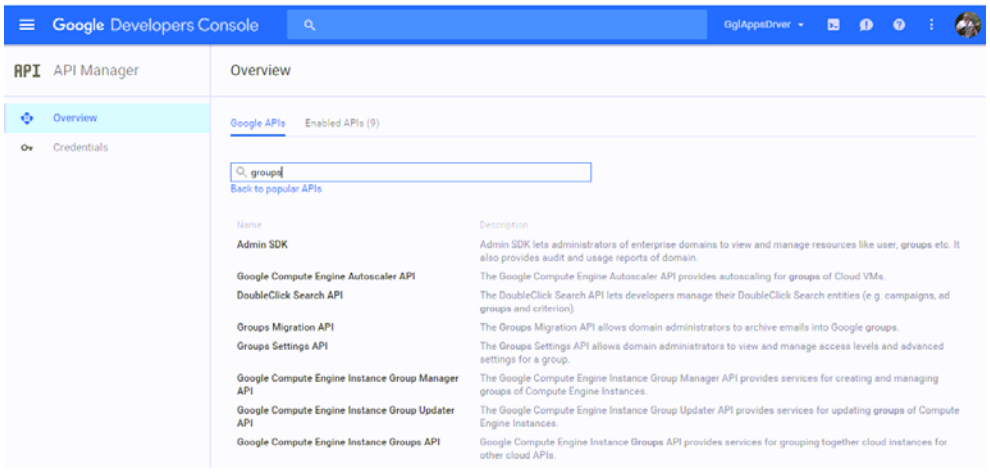
11 Select **Contacts API** from **Google Apps API**



12 Click on **Overview** to return to the list of Google APIs

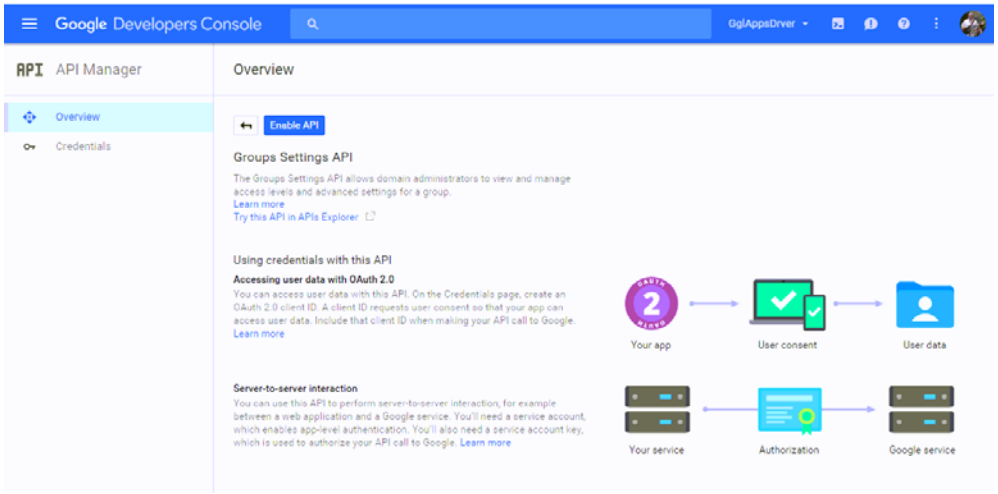
13 Click on **Enable API**

14 Search for the Groups Settings API by typing `Groups` in the **Search all 100+ APIs** control.



15 Select **Groups Settings API** from the list of results.

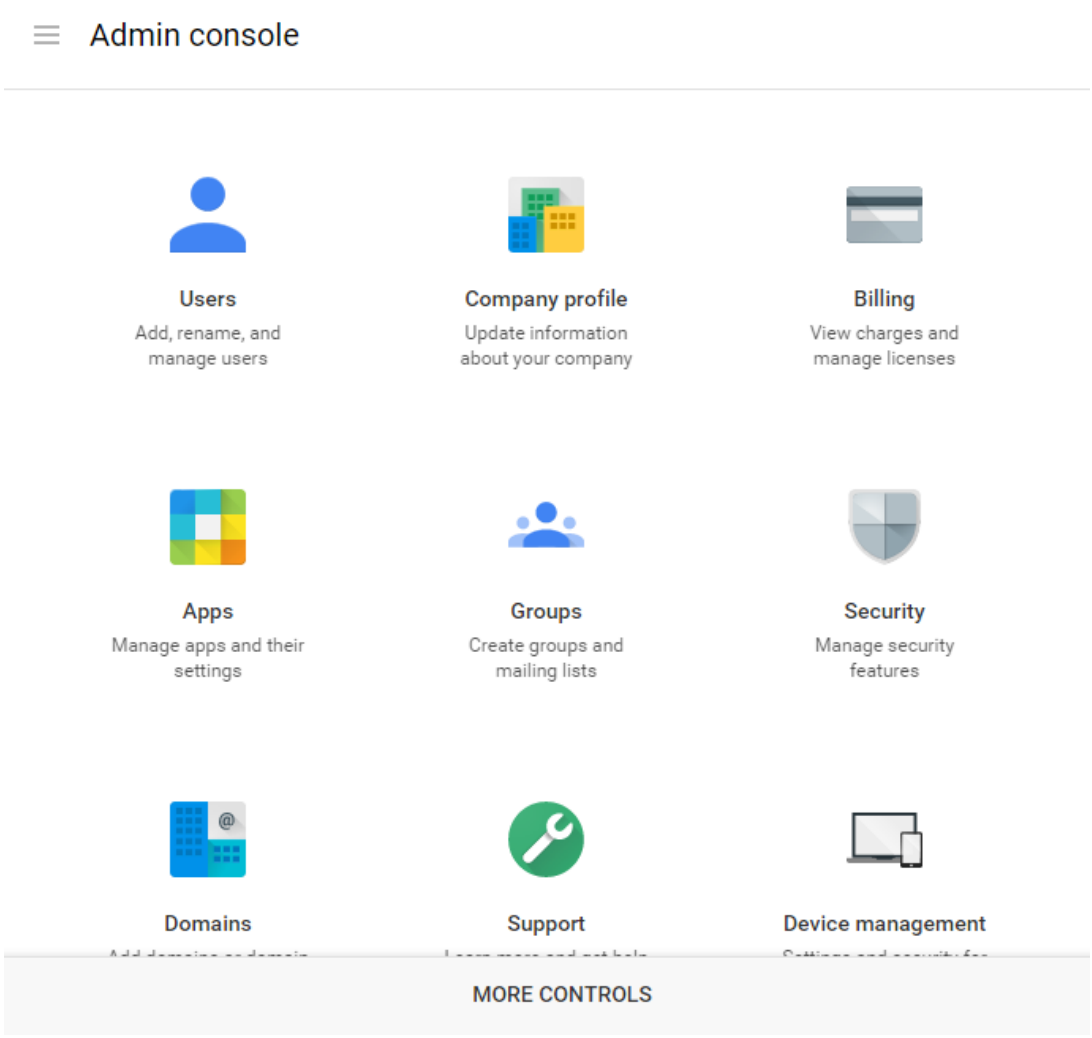
16 Click on **Enable API**



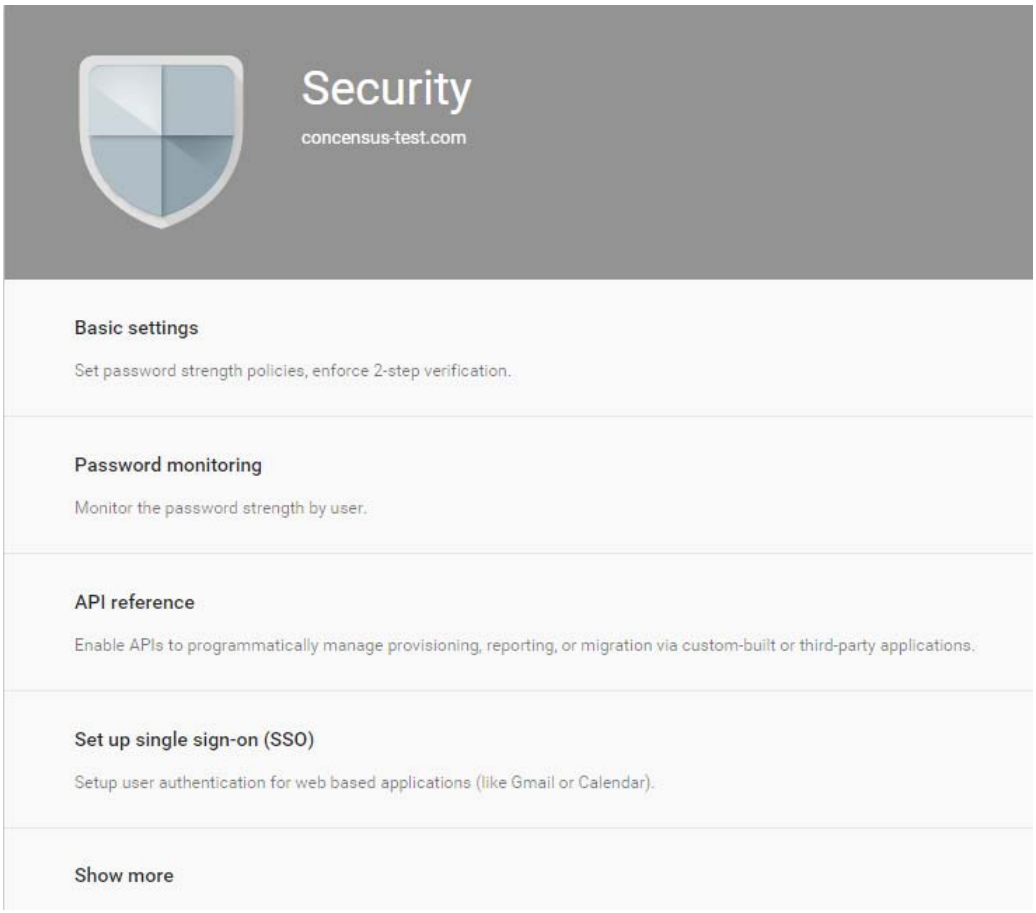
At this point the Service Account Credential to be used by the Google Driver is now created and the APIs required by the Google Driver have been enabled.

3.2.2 Delegate Domain-wide Administrative rights to the Google Service Account

1 Go to the Google Administrative Console

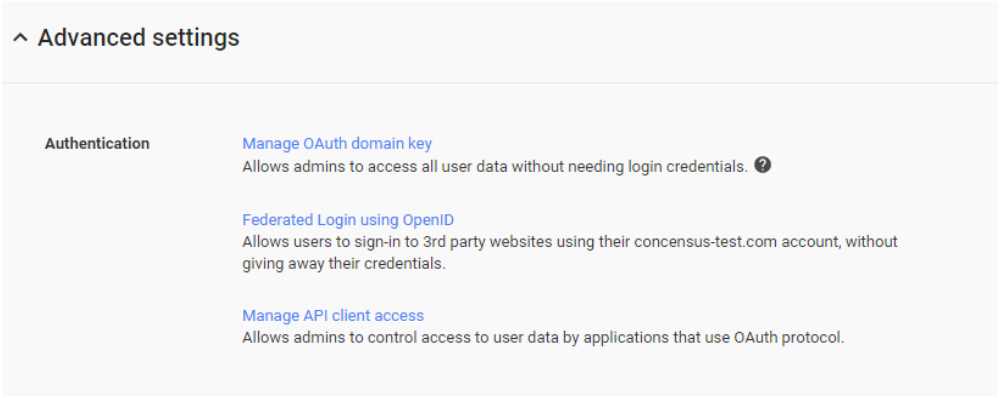


2 Click on the Security icon

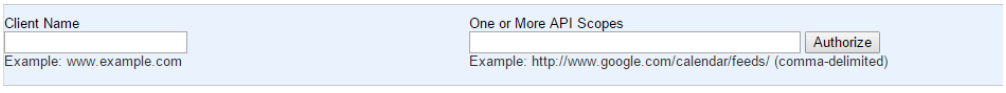


3 Click **Advanced Settings.** If **Advanced Settings** isn't visible, click **Show More**

4 In the Advanced Settings tab, click **Manage API client access** under the Authentication tab.



5 Enter the value for the Client ID from the Service Account Credential in the Developer Console in the **Client Name** field.



6 Enter the list of scopes to authorize for the driver. The list of scopes shown below may not match the driver you are installing. Please refer to the list of scopes that is provided with the driver files in `Directory Scopes.txt` that comes in the Google Apps Driver download package.

https://www.googleapis.com/auth/admin.directory.group,

https://www.googleapis.com/auth/admin.directory.group.member,

https://www.googleapis.com/auth/admin.directory.orgunit,

https://www.googleapis.com/auth/admin.directory.user,

https://www.googleapis.com/auth/admin.directory.user.alias,

https://www.googleapis.com/auth/admin.directory.user.security,

https://www.googleapis.com/auth/admin.directory.userschema,

https://www.googleapis.com/auth/admin.directory.userschema.readonly,

https://www.googleapis.com/auth/userinfo.profile,

https://www.googleapis.com/auth/userinfo.email,

http://www.google.com/m8/feeds,

https://www.googleapis.com/auth/contacts.readonly,

https://www.googleapis.com/auth/apps.groups.settings,

https://apps-apis.google.com/a/feeds/emailsettings/2.0/

3.3 Creating the Driver in Designer

You create the Google Apps driver by importing the driver's configuration file and then modifying the configuration to suit your environment. After you have created and configured the driver, you need to start it.

- Section 3.3.1, "Installing the Current Driver Packages," on page 33
- Section 3.3.2, "Installing the Driver Packages," on page 34
- Section 3.3.3, "Configuring the Driver," on page 40
- Section 3.3.4, "Deploying the Driver," on page 41
- Section 3.3.5, "Starting the Driver," on page 41

3.3.1 Installing the Current Driver Packages

The driver packages contain the items required to create a driver, such as policies, entitlements, filters, and Schema Mapping policies. These packages are only available in Designer and can be updated after they are initially installed. You must have the most current version of the packages in the Package Catalog before you can create a new driver object.

To verify that you have the most recent version of the driver packages in the Package Catalog:

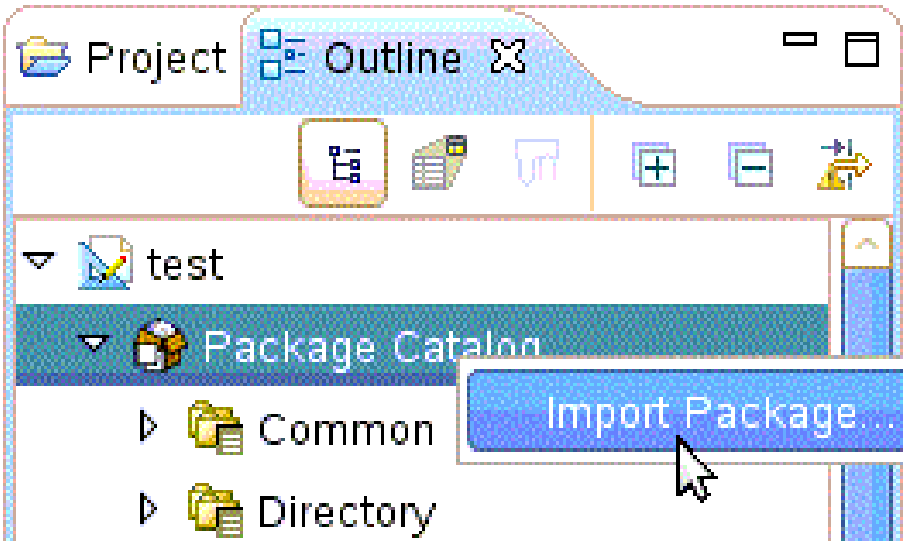
1 Open Designer

2 In the toolbar, Left Click Help > Check for Package Updates

3 Left Click OK to update the packages or Left Click OK if the packages are up-to-date

4 In the Outline view, Right Click the Package Catalog

5 Left Click Import Package



6 Select any Google Apps driver packages

Or

Left Click Select All to import all of the packages displayed.

NOTE: By default, only the base packages are displayed. Deselect Show Base Packages Only to display all packages.

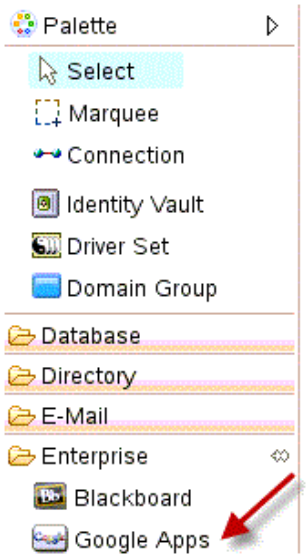
7 Click OK to import the selected packages, and then click OK in the successfully imported packages message.

8 After the current packages are imported, then continue with section,Section 3.3.2, “Installing the Driver Packages,” on page 34

3.3.2 Installing the Driver Packages

1 In Designer, open your project.

2 From the Palette, drag-and-drop the Google Apps driver to the desired driver set in the Modeler.



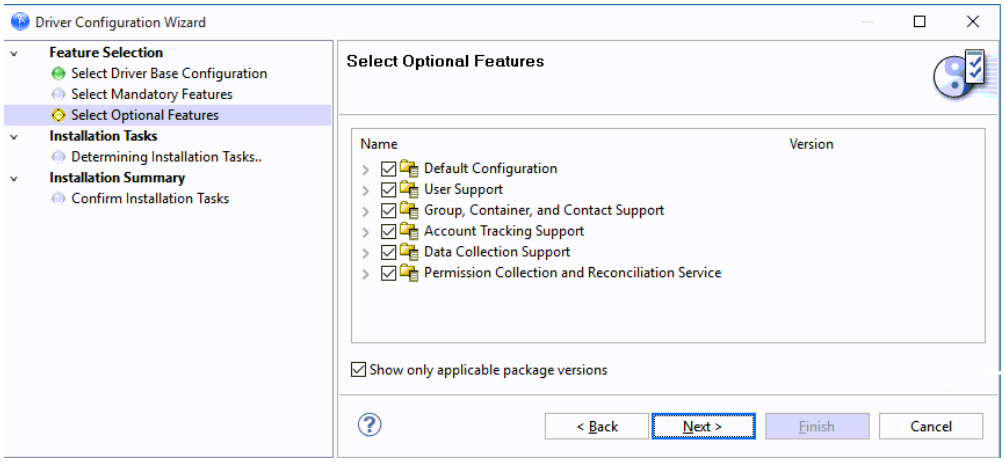
3 Select Google Apps Base, and then Left Click next.

4 Select the optional features to install for the Google Apps driver.

NOTE: By default “show Only applicable packages versions” will be selected as expected.

The Options are:

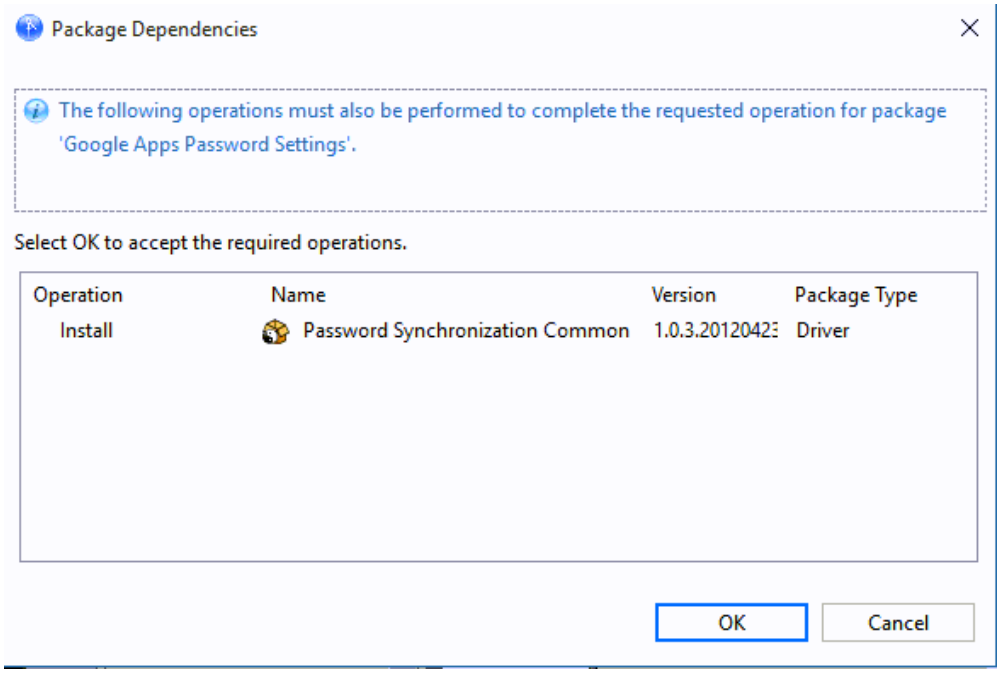
- Google Apps User Package
- Google Apps Organizational Units Package
- Google Apps Groups Package
- Google Apps Contact Package
- Google Apps Account Tracking
- Google Apps Managed System Settings



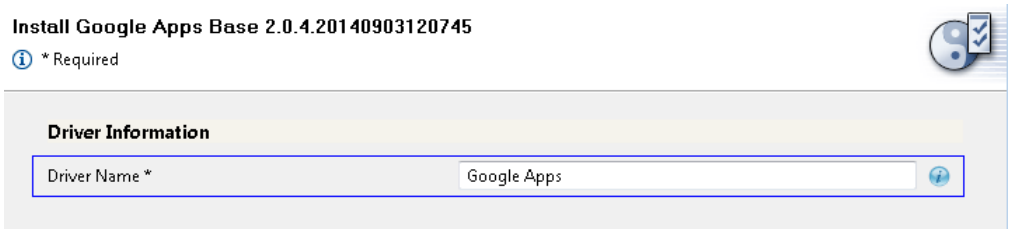
5 Left Click Next

6 (Conditional) If there are package dependencies for the packages you selected to install, you must install them to install the selected package. Left Click OK to install Package Dependencies.

NOTE: There will be mutable instance of this; one for each option selected.



7 On the “Install Google Apps Base” page, specify a name for the driver that is unique within the driver set, and then click next.



8 Configure the authentication of the application.

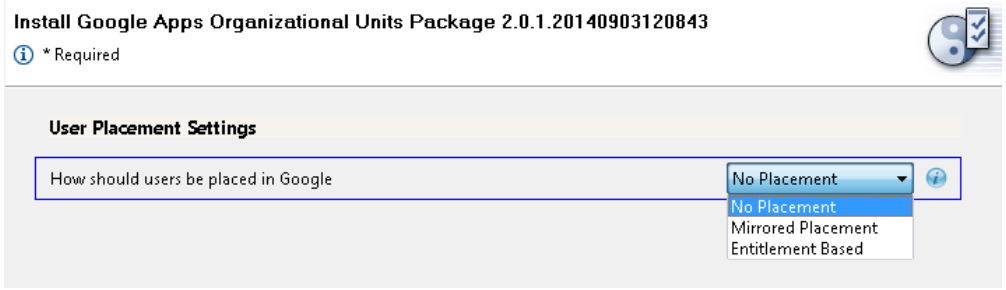
- **Google Apps Domain Name:** Specify the Google Apps Primary Domain Name. (example- yourcompany.com)
- **Google Apps Administrative ID:** Specify the email address of a Google Apps administrator.
- **Password:** Specify the password of the account referenced. Select Next when finished.

9 (Optional) Remote loader configuration: Complete this section if and only if a remote loader is being used.

10 (Optional) Verify Realm information, then select Next.

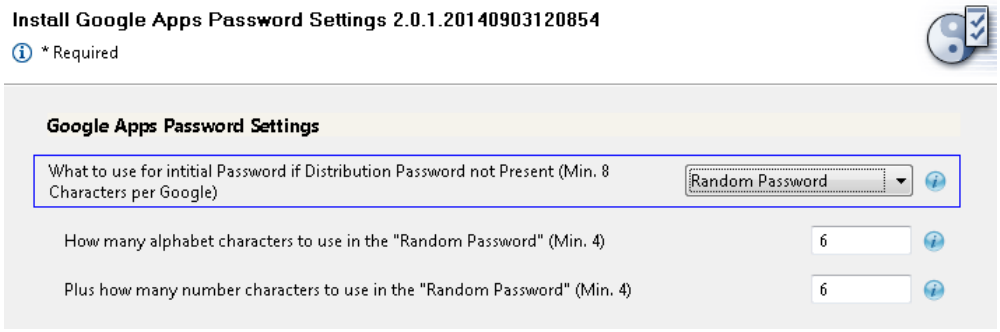
11 (Optional) Specify the name of the Primary Google Apps domain managed by the driver.

12 (Optional) “Installing Google Apps Organizational Units package.” This will configure the placement of users.



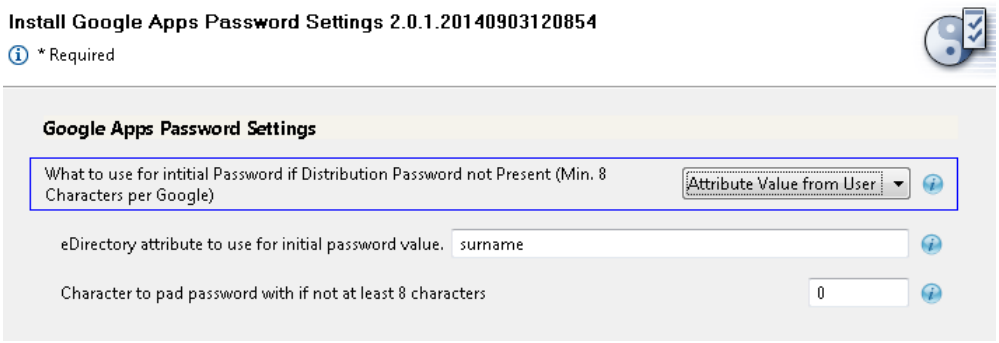
1. **No Placement:** All user accounts will show up in the base of the domain in the Google Management Interface.
2. **Mirror Placement:** The starting base container for all OUs are synchronized to Google and the user’s dn will match from that point forward.
3. **Entitlement Based:** Allows you to select the container in Google that a user will be placed in. It will also grant the location with an Entitlement using RBPMS or Legacy.

13 (Optional) Install Google Apps Password Settings - Random Selected.



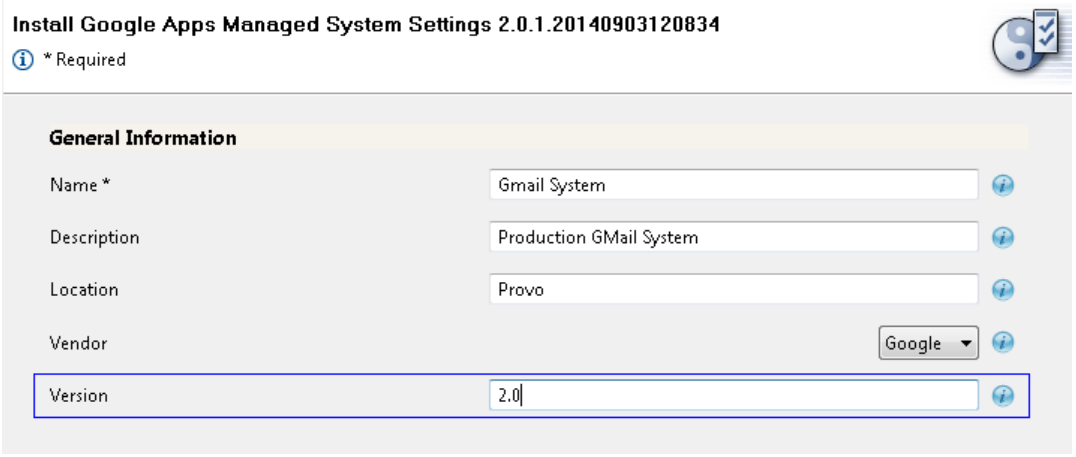
- **Initial Password:** If the system is not set up for Universal Password syncronization or if the user doesn’t have a password set, this will determine the password.
- **Number of Alphabetic Characters:** This determines the number of letters in the random password. This will be combined with the number selected for “number characters”.
- **Number of Number Characters:** This determines the number of number characters in the random password. This will be combined with the number selected for “alphabetic characters”. (Example: if the number 6 is selected for both numbers and letters, a random password will have a length of 12.)

14 (Optional) Install Google Apps Password Settings - Attribute



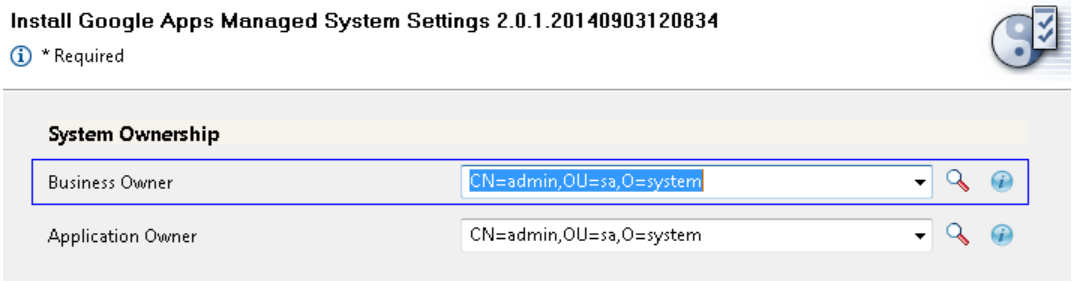
- **eDirectory Attribute:** Enter the name of the attribute in eDirectory that the Google Driver will use for the initial password.
- **Character to pad:** Enter the value to be added to the end of the password if the length of the specified attribute value is less than the minimum number of characters.

15 (Optional) “Installing Google Apps Managed System Setting”



1. **Name:** Specify a descriptive name for the managed system.
2. **Description:** Specify a brief description of the managed system.
3. **Location:** Specify the location of the managed system.
4. **Vendor:** Specify the Vendor of the managed system.
5. **Version:** Specify the version of the managed system.

16 (Optional) Install Google Apps Managed System Settings - System Ownership.



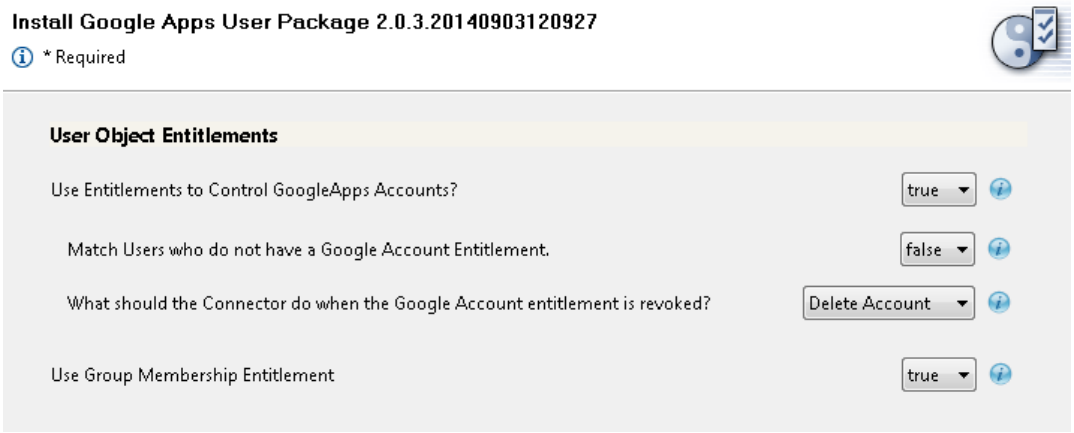
NOTE: Select the Search icon and enter login information to browse to selections.

- **Business Owner:** Specify the business owner of the managed system. Select a user object (not a role, group or container).
- **Application Owner:** Specify the application owner of the managed system. Select a user object (not a role, group or container).

17 (Optional) Install Google Apps Managed System Settings - System Classification.

- **Classification:** Specify one of the following: Mission Critical, Vital, Not Critical, or Other.
- **Environment:** Specify one of the following: Development, Test, Staging, Production, or Other.

18 Install Google Apps User Package



- **Use Entitlements to control Google Apps accounts?** Select either True or False. If set to true, then the entitlement connector must be installed and entitlement must be set to create users in Google Apps.
  - **Match users who do not have a Google account entitlement.** When set to True, users that have not been given an entitlement will be matched to Google users. When set to False, the connector will not attempt to match users without a Google user entitlement and will be blocked at the matching rule.
  - **What should the Connector do when the Google Account entitlement is revoked?** You can choose the default behavior from **Do Nothing**, **Disable Account**, or **Delete Account**
- **Membership Entitlement** Select either True or False.

19 Review the Summary.

20 Select **Finish**.

3.3.3 Configuring the Driver

After importing the driver configuration file, you need to configure the driver before it can run. You should complete the following tasks to configure the driver:

- **Configure the driver properties:** There are many settings that can help you customize and optimize the driver. The settings are divided into categories such as Driver Configuration, Engine Control Values, and Global Configuration Values (GCVs). Although it is important for you to understand all of the settings, your first priority should be to review the Driver Parameters located on the Driver Configuration page. The Driver Parameters and the Global Configuration Values let you configure the Google Apps login information and security credentials, and other parameters associated with the Publisher channel. These settings must be configured properly for the driver to start and function correctly. If you do not have the Driver Properties page displayed in Designer:

1 Open your project.

2 In the Modeler, right-click the driver icon or the driver connection, then select Properties.

3 Make any desired changes, then click OK to save the changes.

4 After the driver is created in Designer, it must be deployed to the Identity Vault. Proceed to Section 3.3.4, “Deploying the Driver,” on page 41

- **Authentication:** This panel contains the user account and connection details for your Google Apps subscription. It also contains additional Remote Loader configuration. The driver will require an account with Google Apps which is an administrator for your Google Apps subscription. It is recommended that a new account be created in your Google Apps domain specifically for this purpose. Make sure that this new account is set to administer your Google Apps domain. These values are set during the default import of the driver.

Google Apps Driver Properties

Property	Description	Example Value
Authentication ID	Google Apps Admin Account	idm@yourdomain.com
Connection Information	Your Google Apps Domain	yourdomain.com

Be sure to set the account password in the Application Authentication section of the driver properties.

Driver Configuration

- **Configure the driver parameters:** The driver parameters panel contains driver-specific configuration.
  1. **Driver Options** The Google Apps driver does not use any Driver Options. This panel is intentionally blank.
  2. **Subscriber Options:**
     - **Hash Password** Select **True** to have the Google driver apply an MD5 hash to passwords prior to sending them to Google.
  3. **Publisher Options:**
     - **Heartbeat Interval:** Specify the length of time in seconds between heartbeats emitted by the Google driver’s publisher channel.

If this GCV is set to true then Groups that have not been given a Google Group Create entitlement will be matched to existing Google Groups. Otherwise the connector will not attempt to match Groups without a Google Group Create entitlement they will just be blocked at the matching rule.

- **Global Configuration Values (GCVs)**

  The GCVs are defined in Table A-6 on page 77

After completing the configuration tasks, continue with Section 3.3.4, “Deploying the Driver,” on page 41.

3.3.4 Deploying the Driver

After the driver is created in Designer, it must be deployed into the Identity Vault.

1. In Designer, open your project.
2. In the Modeler, right-click the driver icon or the driver connection, then select Live > Deploy.
3. Read through the deployment summary, and then click Deploy.
4. Read the success message, and then click OK.
5. Click Define Security Equivalence to assign rights to the driver.

   The driver requires rights to objects within the Identity Vault. The Admin user object is most often used to supply these rights. However, you might want to create a DriversUser (for example) and assign security equivalence to that user. Any Rights the driver needs to have on the server need to be assigned to the DriversUser object.

   5a Click Add, then browse to and select the object with the correct rights.

   5b Click OK twice.
6. Click Exclude Administrative Roles to exclude users that should not be synchronized.

   6a Click Add, then browse to and select the user object you want to exclude.

   6b Click OK.

   6c Repeat Step 6a and 6b for each object you want to exclude.

   6d Click OK.
7. Click OK

3.3.5 Starting the Driver

When a driver is created, it is stopped by default. To make the driver work, you must start the driver. Identity Manager is an event-driven system, so after the driver is started, it won’t do anything until an event occurs.

To start the driver:

1. In Designer, select the project view.
2. Click on the Google Apps driver.
3. Click the green start icon.

3.4 Activating the Driver

If you created the Google Apps driver in a driver set that has not been activeate, you must activate the driver with a Google Apps Driver activation within 90 days. If you do not apply a Google Apps Driver activation within 90 days, the driver will stop working.

For more information on activation, refer to “Activating Novell Identity Manager Products” in the Identity Manager 4.0 Framework Installation Guide.

The drivers that are included in the Integration Module for Tools are:

- Driver for Delimited Text
- Driver for SOAP

For information on activation, refer to “Activating Novell Identity Manager Products” in the *Identity Manager 4.0.2 Framework Installation Guide*.

3.5 Google Apps Requirements

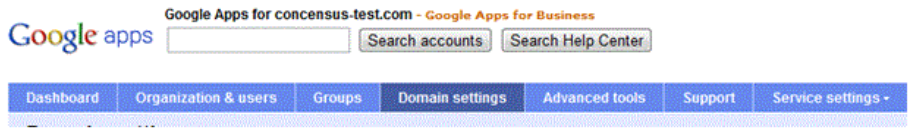
In order for the driver to interact with your domain, the following steps are required:

3.5.1 Enabling the Google Provisioning API Access

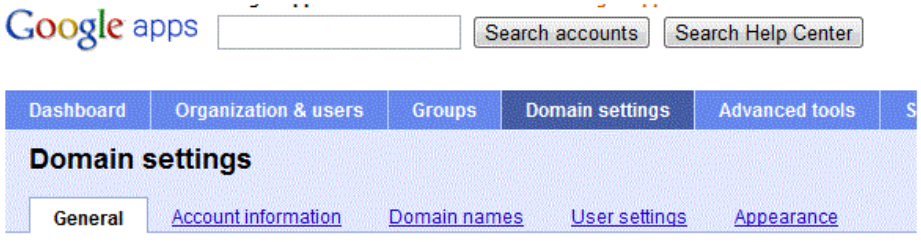
The driver will provision users into Google Apps for Business or Google Apps for Education edition services. It is necessary to enable the Google Provisioning API of your Google Apps subscription before the driver can interoperate with Google Apps.

To enable Google’s API access:

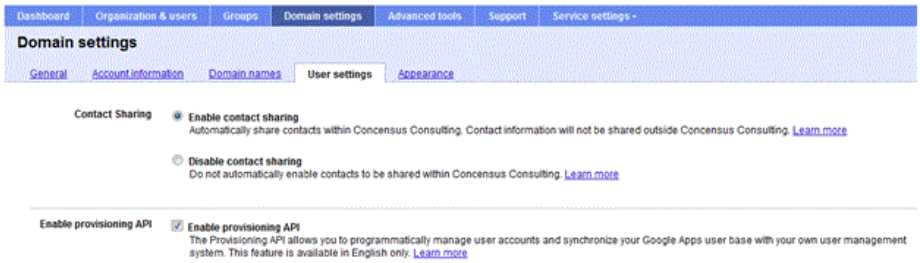
1. Using a web browser, log into the Google Apps Administration Console, typically found at http://www.google.com/a/yourdomainname, where yourdomainname is the Google Apps domain for your subscription. For example, if your Google Apps accounts take the form of username@mydomain.com, then your domain name is mydomain.com.
2. From the Dashboard, select "Domain Settings".



3. From the Domain Settings management page, select "User Settings".



4. Scroll down the Settings page and check the box labeled "Enable Provisioning API".



5 Save the settings by clicking the "Save Changes" button at the bottom of the page.

You can confirm that the API has been enabled by clicking the "Organizations and Users" button at the top of the management console.

This enables the Provisioning API interface for your Google Apps subscription. This interface provides the access methods which the driver will use to provision and manage users and groups in Google Apps.

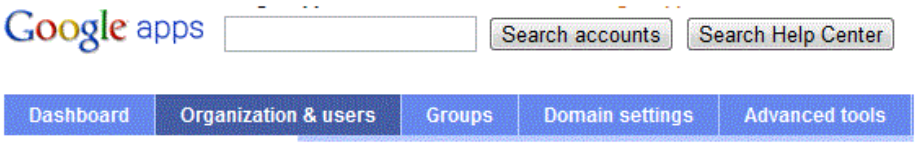
3.5.2 Creating a Google Administrative Account

In order for the Google Driver to access the Google Domain and perform administrative functions such as creating users, the driver must log in to the domain using a Google account with Administrative Privileges.

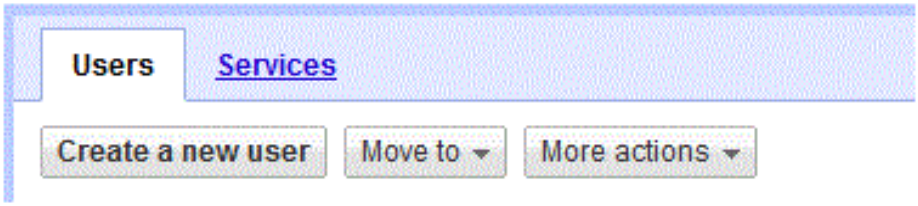
To access the Google Domain:

1 Using a web browser, log into the Google Apps Administration Console, typically found at http://www.google.com/a/yourdomainname, where yourdomainname is the Google Apps domain for your subscription. For example, if your Google Apps accounts take the form of username@mydomain.com, then your domain name is mydomain.com.

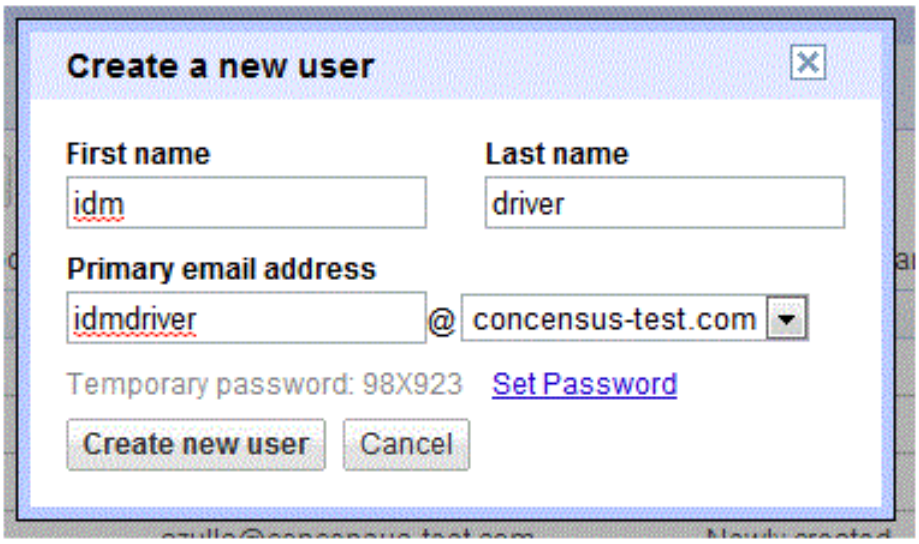
2 From the Dashboard, select "Organization & Users".



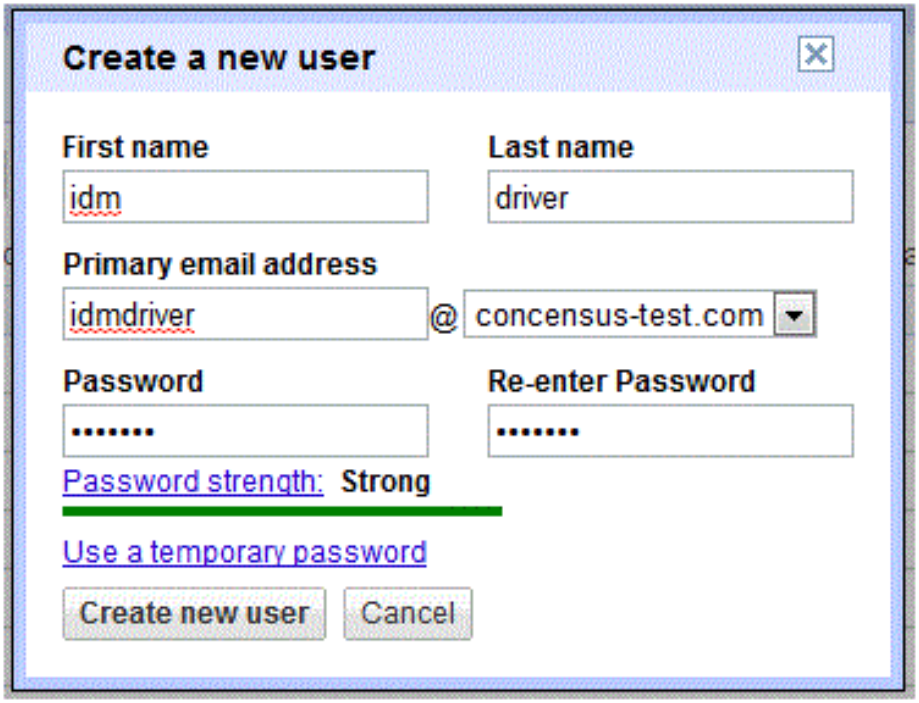
3 Click the Create a New User button.



4 Enter a First Name, Last Name and email address.



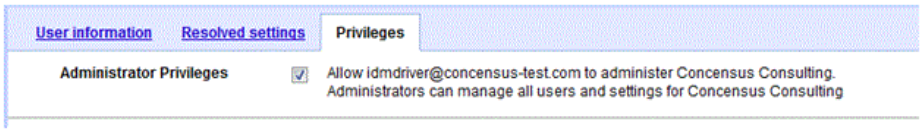
5 Click on the Set Password link and set the password you desire to user for the driver ID.



6 Click Create new user.

7 Find your new Driver ID in the list of Users and select it.

8 Click on the Privileges tab and check the Administrator Privileges box and click Save Changes



9 Log out of the Google console and log back in using the new Driver ID.

10 Accept the Google Terms of Service.

Now this ID can be used by the driver to manage the Google domain.

4 Upgrading an Existing Driver

The following sections provide information to help you upgrade an existing driver:

4.1 Supported Upgrade Paths

You can upgrade from any 4.0.x version of the Google driver. Upgrading a pre 4.0.x version of the driver directly to version 4.1.1 or later is not supported. Google Driver prior to version 4.0.x used the Provisioning API. Google ended support for that API in June of 2015.

4.2 What’s New in Version 4.1.1.x

- Support for Directory API.
- All Google APIs Authenticate using OAuth2.
- Support for Custom Schema fields in Google Apps.
- Enhanced handling of Google Limits exceptions using exponential back-off algorithms.
- Support for Alias attributes on user create.

4.3 Upgrade an Existing Driver

4.3.1 Instructions for Patching from Google Apps Driver v 4.0.x

Linux

1. The connector binary and accessory jar files are located in the eDirectory DirXML class library path. This path is dependent on your install location. On Linux hosts, it is typically `opt/novell/eDirectory/lib/dirxml/classes`.
2. Update the GoogleApps Driver Packages
   - Google Apps Account Tracking
   - Google Apps Base
   - Google Apps Configuration
   - Google Apps Contact Package
   - Google Apps Entitlements
   - Google Apps Groups Package
   - Google Apps Managed System Settings
   - Google Apps Organizational Unites Package

- Google Apps Password Settings
- Google Apps User Package

3. The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.
4. Stop all drivers
5. Stop eDirectory
6. Delete the existing Google Apps driver binary. The files to delete are:
   - gmailshim.jar
   - `google-api-client-1.19.1.jar`
   - `google-api-client-java6-1.19.1.jar`
   - `google-http-client-1.19.0.jar`
   - `google-http-client-gson-1.19.0.jar`
   - `google-http-client-jackson2-1.19.0.jar`
   - `google-oauth-client-1.19.0.jar`
   - `google-oauth-client-java6-1.19.0.jar`
   - `google-oauth-client-jetty-1.19.0.jar`
   - `google-api-services-admin-directory_v1-rev50-1.19.1.jar`
   - `google-api-services-oauth2-v2-rev87-1.19.1.jar`
7. Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.
   - `gmailshim.jar`
   - `google-api-client-1.20.1.jar`
   - `google-api-client-java6-1.20.1.jar`
   - `google-http-client-1.20.0.jar`
   - `google-http-client-gson-1.20.0.jar`
   - `google-http-client-jackson2-1.20.0.jar`
   - `google-oauth-client-1.20.0.jar`
   - `google-oauth-client-java6-1.20.0.jar`
   - `google-oauth-client-jetty-1.20.0.jar`
   - `google-api-services-admin-directory_v1-rev55-1.20.0.jar`
   - `google-api-services-oauth2-v1-rev95-1.20.0.jar`
   - `google-api-services-groupssettings-v1-rev54-1.20.0.jar`
8. Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)
9. Enable the Groups Settings API in the Google Developers Console. Section 3.1.2, “Enabling Google API Access,” on page 18
10. Re-authorize the Client ID for your Service Account Credential using the scopes provided in `DirectoryScopes.txt` provided with the patch.Section 3.2.2, “Delegate Domain-wide Administrative rights to the Google Service Account,” on page 30

Windows

1. The connector binary and accessory jar files are located in the eDirectory DirXML class library path. This path is dependent on your install location. On Windows hosts, it is typically `[Install_Location]\NDS\lib\.`
2. Update the GoogleApps Driver Packages
   - Google Apps Account Tracking
   - Google Apps Base
   - Google Apps Configuration
   - Google Apps Contact Package
   - Google Apps Entitlements
   - Google Apps Groups Package
   - Google Apps Managed System Settings
   - Google Apps Organizational Unites Package
   - Google Apps Password Settings
   - Google Apps User Package
3. The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.
4. Stop all drivers
5. Stop eDirectory
6. Delete the existing Google Apps driver binary. The files to delete are:
   - gmailshim.jar
   - `google-api-client-1.19.1.jar`
   - `google-api-client-java6-1.19.1.jar`
   - `google-http-client-1.19.0.jar`
   - `google-http-client-gson-1.19.0.jar`
   - `google-http-client-jackson2-1.19.0.jar`
   - `google-oauth-client-1.19.0.jar`
   - `google-oauth-client-java6-1.19.0.jar`
   - `google-oauth-client-jetty-1.19.0.jar`
   - `google-api-services-admin-directory_v1-rev50-1.19.1.jar`
   - `google-api-services-oauth2-v2-rev87-1.19.1.jar`
7. Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.
   - `gmailshim.jar`
   - `google-api-client-1.20.1.jar`
   - `google-api-client-java6-1.20.1.jar`
   - `google-http-client-1.20.0.jar`
   - `google-http-client-gson-1.20.0.jar`
   - `google-http-client-jackson2-1.20.0.jar`
   - `google-oauth-client-1.20.0.jar`
   - `google-oauth-client-java6-1.20.0.jar`

- `google-oauth-client-jetty-1.20.0.jar`
- `google-api-services-admin-directory_v1-rev55-1.20.0.jar`
- `google-api-services-oauth2-v1-rev95-1.20.0.jar`
- `google-api-services-groupssettings-v1-rev54-1.20.0.jar`

8. Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)
9. Enable the Groups Settings API in the Google Developers Console. Section 3.1.2, “Enabling Google API Access,” on page 18
10. Re-authorize the Client ID for your Service Account Credential using the scopes provided in `DirectoryScopes.txt` provided with the patch.Section 3.2.2, “Delegate Domain-wide Administrative rights to the Google Service Account,” on page 30

Remote Loader

1. Remote loader driver paths are dependent on how the remote loader is installed. Locate the existing gmailshim.jar on the remote loader host to identify the correct path.
2. Update the GoogleApps Driver Packages
   - Google Apps Account Tracking
   - Google Apps Base
   - Google Apps Configuration
   - Google Apps Contact Package
   - Google Apps Entitlements
   - Google Apps Groups Package
   - Google Apps Managed System Settings
   - Google Apps Organizational Unites Package
   - Google Apps Password Settings
   - Google Apps User Package
3. The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.
4. Stop all drivers
5. Stop eDirectory
6. Delete the existing Google Apps driver binary. The files to delete are:
   - gmailshim.jar
   - `google-api-client-1.19.1.jar`
   - `google-api-client-java6-1.19.1.jar`
   - `google-http-client-1.19.0.jar`
   - `google-http-client-gson-1.19.0.jar`
   - `google-http-client-jackson2-1.19.0.jar`
   - `google-oauth-client-1.19.0.jar`
   - `google-oauth-client-java6-1.19.0.jar`
   - `google-oauth-client-jetty-1.19.0.jar`
   - `google-api-services-admin-directory_v1-rev50-1.19.1.jar`
   - `google-api-services-oauth2-v2-rev87-1.19.1.jar`

7 Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.

- `gmailshim.jar`
- `google-api-client-1.20.1.jar`
- `google-api-client-java6-1.20.1.jar`
- `google-http-client-1.20.0.jar`
- `google-http-client-gson-1.20.0.jar`
- `google-http-client-jackson2-1.20.0.jar`
- `google-oauth-client-1.20.0.jar`
- `google-oauth-client-java6-1.20.0.jar`
- `google-oauth-client-jetty-1.20.0.jar`
- `google-api-services-admin-directory_v1-rev55-1.20.0.jar`
- `google-api-services-oauth2-v1-rev95-1.20.0.jar`
- `google-api-services-groupssettings-v1-rev54-1.20.0.jar`

8 Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

9 Enable the Groups Settings API in the Google Developers Console. Section 3.1.2, “Enabling Google API Access,” on page 18

10 Re-authorize the Client ID for your Service Account Credential using the scopes provided in `DirectoryScopes.txt` provided with the patch.Section 3.2.2, “Delegate Domain-wide Administrative rights to the Google Service Account,” on page 30

4.3.2 Instructions for Patching from Google Apps Driver v 4.1.0

Linux

1 The connector binary and accessory jar files are located in the eDirectory DirXML class library path. This path is dependent on your install location. On Linux hosts, it is typically `opt/novell/eDirectory/lib/dirxml/classes`.

2 Update the GoogleApps Driver Packages

- Google Apps Account Tracking
- Google Apps Base
- Google Apps Configuration
- Google Apps Contact Package
- Google Apps Entitlements
- Google Apps Groups Package
- Google Apps Managed System Settings
- Google Apps Organizational Unites Package
- Google Apps Password Settings
- Google Apps User Package

3 The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.

4 Stop all drivers

5 Stop eDirectory

6 Delete the existing Google Apps driver binary. The files to delete are:

- gmailshim.jar
- `google-api-client-1.19.1.jar`
- `google-api-services-admin-directory_v1-rev53-1.20.0.jar`
- `google-api-services-oauth2-v2-rev87-1.19.1.jar`

7 Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.

- `gmailshim.jar`
- `google-api-client-1.20.1.jar`
- `google-api-services-admin-directory_v1-rev55-1.20.0.jar`
- `google-api-services-oauth2-v1-rev95-1.20.0.jar`

8 Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

Windows

1 The connector binary and accessory jar files are located in the eDirectory DirXML class library path. This path is dependent on your install location. On Windows hosts, it is typically `[Install_Location]\NDS\lib\.`

2 Update the GoogleApps Driver Packages

- Google Apps Account Tracking
- Google Apps Base
- Google Apps Configuration
- Google Apps Contact Package
- Google Apps Entitlements
- Google Apps Groups Package
- Google Apps Managed System Settings
- Google Apps Organizational Unites Package
- Google Apps Password Settings
- Google Apps User Package

3 The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.

4 Stop all drivers

5 Stop eDirectory

6 Delete the existing Google Apps driver binary. The files to delete are:

- gmailshim.jar
- `google-api-client-1.19.1.jar`
- `google-api-services-admin-directory_v1-rev53-1.20.0.jar`
- `google-api-services-oauth2-v2-rev87-1.19.1.jar`

7 Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.

- `gmailshim.jar`
- `google-api-client-1.20.1.jar`

- `google-api-services-admin-directory_v1-rev55-1.20.0.jar`
- `google-api-services-oauth2-v1-rev95-1.20.0.jar`

8 Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

Remote Loader

1 Remote loader driver paths are dependent on how the remote loader is installed. Locate the existing gmailshim.jar on the remote loader host to identify the correct path.

2 Update the GoogleApps Driver Packages

- Google Apps Account Tracking
- Google Apps Base
- Google Apps Configuration
- Google Apps Contact Package
- Google Apps Entitlements
- Google Apps Groups Package
- Google Apps Managed System Settings
- Google Apps Organizational Unites Package
- Google Apps Password Settings
- Google Apps User Package

3 The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.

4 Stop all drivers

5 Stop eDirectory

6 Delete the existing Google Apps driver binary. The files to delete are:

- gmailshim.jar
- `google-api-client-1.19.1.jar`
- `google-api-services-admin-directory_v1-rev53-1.20.0.jar`
- `google-api-services-oauth2-v2-rev87-1.19.1.jar`

7 Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.

- `gmailshim.jar`
- `google-api-client-1.20.1.jar`
- `google-api-services-admin-directory_v1-rev55-1.20.0.jar`
- `google-api-services-oauth2-v1-rev95-1.20.0.jar`

8 Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

4.3.3 Instructions for Patching from Google Apps Driver v 4.1.0.1 or 4.1.0.2

Linux

1. The connector binary and accessory jar files are located in the eDirectory DirXML class library path. This path is dependent on your install location. On Linux hosts, it is typically `opt/novell/eDirectory/lib/dirxml/classes.`
2. Update the GoogleApps Driver Packages
   - Google Apps Account Tracking
   - Google Apps Base
   - Google Apps Configuration
   - Google Apps Contact Package
   - Google Apps Entitlements
   - Google Apps Groups Package
   - Google Apps Managed System Settings
   - Google Apps Organizational Unites Package
   - Google Apps Password Settings
   - Google Apps User Package
3. The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.
4. Stop all drivers
5. Stop eDirectory
6. Delete the existing Google Apps driver binary. The files to delete are:
   - gmailshim.jar
   - `google-api-client-1.19.1.jar`
7. Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.
   - `gmailshim.jar`
   - `google-api-client-1.20.1.jar`
8. Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

Windows

1. The connector binary and accessory jar files are located in the eDirectory DirXML class library path. This path is dependent on your install location. On Windows hosts, it is typically `[Install_Location]\NDS\lib\.`
2. Update the GoogleApps Driver Packages
   - Google Apps Account Tracking
   - Google Apps Base
   - Google Apps Configuration
   - Google Apps Contact Package
   - Google Apps Entitlements

- Google Apps Groups Package
- Google Apps Managed System Settings
- Google Apps Organizational Unites Package
- Google Apps Password Settings
- Google Apps User Package

3 The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.

4 Stop all drivers

5 Stop eDirectory

6 Delete the existing Google Apps driver binary. The files to delete are:

- gmailshim.jar
- `google-api-client-1.19.1.jar`

7 Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.

- `gmailshim.jar`
- `google-api-client-1.20.1.jar`

8 Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

Remote Loader

1 Remote loader driver paths are dependent on how the remote loader is installed. Locate the existing gmailshim.jar on the remote loader host to identify the correct path.

2 Update the GoogleApps Driver Packages

- Google Apps Account Tracking
- Google Apps Base
- Google Apps Configuration
- Google Apps Contact Package
- Google Apps Entitlements
- Google Apps Groups Package
- Google Apps Managed System Settings
- Google Apps Organizational Unites Package
- Google Apps Password Settings
- Google Apps User Package

3 The Google Apps driver packages can be used to import a new driver or perform an in-place update of an existing driver.

4 Stop all drivers

5 Stop eDirectory

6 Delete the existing Google Apps driver binary. The files to delete are:

- gmailshim.jar
- `google-api-client-1.19.1.jar`

7 Copy the driver binary files provided with the patch to the eDirectory DirXML Library Path.

- `gmailshim.jar`
- `google-api-client-1.20.1.jar`

8 Restart eDirectory - `rcndsd restart` (all drivers marked as automatic start will restart)

5 Customizing the Driver

The following sections provide information to help you understand what the driver does and what customization you might need to make to the driver:

5.1 Managing the Driver

As you work with the Google Apps driver, there are a variety of management tasks you might need to perform, including the following:

- Starting, stopping, and restarting the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver’s health status
- Backing up the driver
- Inspecting the driver’s cache files
- Viewing the driver’s statistics
- Using the DirXML Command Line utility to perform management tasks through scripts
- Securing the driver and its information

Because these tasks, as well as several others, are common to all Identity Manager drivers, they are included in one reference, the *Identity Manager 4.0.2 Upgrade and Migration Guide*.

5.2 Schema Mapping

This section details the default schema mapping of the driver. The schema map details how Identity Vault attributes and classes are translated into Google Apps attributes and classes.

The section includes:

5.2.1 User Attributes Mapping

Identity Vault	Google Apps
User	UserEntry

Identity Vault	Google Apps
	Agreed to terms
preferredName	Alias
assistant	Assistant
assistantPhone	AssistantPhoneNumber
	Brother
	CallbackPhoneNumber
	CarPhoneNumber
	ChangePasswordAtNextLogin
	Child
	CompanyMainPhoneNumber
	DomesticPartner
	ExternalId
Surname	FamilyName
	Father
	Friend
	GeneralPhoneNumber
Given Name	GivenName
	GmailSettingsDelegates
	GmailSettingsEnableIMAP
	GmailSettingsEnablePOP
	GmailSettingsForwarding
	GmailSettingsLabel
Language	GmailSettingsLanguage
	GmailSettingsSendAs
	GmailSettingsSignature
Groups Memberships	Groups
	Hidden
	HomeCity
	HomeCountry
	HomeCountryCode
	HomeFaxPhoneNumber
	HomeFormattedAddress
	HomePhoneNumber

Identity Vault	Google Apps
	HomePostalCode
	HomeRegion
Home Phone	HomeStreetAddress
	IpWhiteListed
	IsAdmin
internationalISDNNumber	ISDNPhoneNumber
Login Disabled	IsSupended
	MainPhoneNumber
manager	Manager
mobile	MobilePhoneNumber
	Mother
	OrgCostCenter
OU	OrgDepartment
	OrgJobDescription
L	OrgLocation
company	OrgName
	OrgSymbol
Title	OrgTitle
	OtherEmailAddress
	OtherFaxPhoneNumber
otherPhoneNumber	OtherPhoneNumber
Pager	PagerPhoneNumber
	Parent
	Partner
nspmDistributionPassword	Password
	Photo
	RadioPhoneNumber
	ReferredBy
	Sister
	Spouse
TelexNumber	TelexPhoneNumber
	TTY_TDDPhoneNumber
CN	UserName

Identity Vault	Google Apps
	WorkCity
	WorkCountry
	WorkCountryCode
Fascimile Telephone Number	WorkFaxPhoneNumber
	WorkFormattedAddress
	WorkMobilePhoneNumber
	WorkPagerPhoneNumber
	WorkPhoneNumber
	WorkPostalCode
S	WorkRegion
SA	WorkStreetAddress

5.2.2 Group Attribute Mapping

Identity Vault	Google Apps
Group	Group
	AllowExternalMembers
	AllowGoogleCommunication
	AllowWebPosting
	ArchiveOnly
	CustomReplyTo
	DefaultMessageDenyNotificationText
Description	Description
DirXML-GAGroupEmailAddress	EmailAddress
	IncludeInGlobalAddressList
	IsArchived
	MaxMessageBytes
Member	Members
	MembersCanPostAsTheGroup
	MessageDisplayFont
	MessageModerationLevel
CN	Name
Owner	Owners

Identity Vault	Google Apps
	PrimaryLanguage
	ReplyTo
	SendMessageDenyNotification
	ShowInGroupDirectory
	SpamModerationLevel
	WhoCanContactOwner
	WhoCanInvite
	WhoCanJoin
	WhoCanLeaveGroup
	WhoCanPostMessage
	WhoCanViewGroup
	WhoCanViewMembership

5.2.3 Organizational Unit Attribute Mapping

Identity Vault	Google Apps
Organizational Unit	Organizational Unit
	BlockInheritance
Description	Description
OU	Name
	OrgUnitPath
	ParentOrgUnitPath

5.2.4 Contact Attribute Mapping

The ContactEntry class does not map directly to a class in eDirectory. The schema can be extended (or mapped to the user object class). The driver contains a sample GoogleContact.sch file that can be used to extend the eDirectory schema. The following table lists the available attributes within Google Apps.

Identity Vault (EXAMPLE)	Google Apps
GoogleContact	ContactEntry
assistant	Assistant
assistantPhone	AssistantPhoneNumber
	Brother

Identity Vault (EXAMPLE)	Google Apps
	CallbackPhoneNumber
	CarPhoneNumber
	Child
	CompanyMainPhoneNumber
	Cube
	DomesticPartner
	Father
	Friend
	GeneralPhoneNumber
	HomeCity
	HomeCountry
	HomeCountryCode
	HomeEmailAddress
	HomeFaxPhoneNumber
Home Phone	HomePhoneNumber
	HomePostalCode
	HomeRegion
	HomeStreetAddress
internationalISDNNumber	ISDNPhoneNumber
	MainPhoneNumber
Manager	Manager
mobile	MobilePhoneNumber
	Mother
	OfficeMailstop
OU	OrgDepartment
	OrgJobDescription
L	OrgLocation
company	OrgName
	OrgSymbol
Title	OrgTitle
	OtherEmailAddress
	OtherFaxPhoneNumber
otherPhoneNumber	OtherPhoneNumber

Identity Vault (EXAMPLE)	Google Apps
Pager	PagerPhoneNumber
	Parent
	Partner
	ProfileAdditionalName
	ProfileFamilyName
	ProfileFullName
	ProfileGivenName
	ProfileNamePrefix
	ProfileNameSuffix
	RadioPhoneNumber
	ReferredBy
	Sister
	Spouse
TelexNumber	TelexPhoneNumber
	TTY_TDDPhoneNumber
	WorkCity
	WorkCountry
	WorkCountryCode
	WorkEmailAddress
Facsimile Telephone Number	WorkFaxPhoneNumber
mobile	WorkMobilePhoneNumber
pager	WorkPagerPhoneNumber
Telephone Number	WorkPhoneNumber
	WorkPostalCode
	WorkRegion
	WorkStreetAddress

5.2.5 Using Google Custom Schema

The Google Apps Directory API provides the ability to extend the schema of a UserEntry object through the use of Google Custom Schema. Customers can create multiple custom schemas, each of which can define multiple custom attributes. These fields can be used to hold attribute data. Adding Custom Schema effectively extends the application schema managed by the driver. When the driver is asked to refresh application schema from Designer or iManager, the driver queries all of the Custom Schema objects, and adds all of the attributes to the application schema. The custom

schema attributes appear in the schema as <Schema name>.<Attribute Name>. Once the driver has returned the new schema attributes, the attributes are available to be included in the filter, schema mapped, and used in the Policy Builder.

Google Custom Schema attribute definitions carry metadata to indicate whether or not the attribute is multi-valued, as well as the datatype of the field. Google supports the following datatypes:

- BOOL
- DATE
- DOUBLE
- EMAIL
- INT64
- PHONE
- STRING

6 Managing the Driver

As you work with the Google Apps driver, there are several management tasks you might need to perform, including the following:

- Starting, stopping, and restarting the driver
- Viewing driver version information
- Using Named Passwords to securely store passwords associated with the driver
- Monitoring the driver's health status
- Backing up the driver
- Inspecting the driver's cache files
- Viewing the driver's statistics
- Using the DirXML Command Line utility to perform management tasks through scripts
- Securing the driver and its information

Because these tasks, as well as several others, are common to all Identity Manager drivers, they are included in one reference, the *NetIQ Identity Manager 4.0.2 Common Driver Administration Guide*.

7 Troubleshooting the Driver

You can log Identity Manager events by using Novell Event Auditing Service. Using this service in combination with the driver log level setting provides you with tracking control at a very granular level.

This section contains the following information on error messages:

7.1 Reporting Errors to Identity Manager

The driver reports errors occurring in both the driver and the Google Domain. All errors reported by the driver follow the Identity Manager Driver error reporting scheme of Status Level and Status Type.

Status Level	Description
Success	The operation succeeded
Warning	The operation succeeded with a warning
Retry	The operation failed because of an error not related to invalid data or a memory or execution error. These are transient errors. For instance, the Google driver issues a Retry when Google reports a Server Busy error.
Error	The operation failed due to an error in xml formatting or a data error.
Fatal	The operation failed as a result of an unrecoverable condition, such as an OutOfMemoryException.

The Status Type provides a way for a driver to indicate the category of the error. For instance, the driver can use Status Type to indicate if a Retry has been issued as a result of application connectivity error. When handling an exception or an error as a result of a transient condition the driver will disconnect from the Google domain and then send a retry request to the Identity Manager engine. The default retry interval is 30 seconds. Once 30 seconds has elapsed the IDM engine will send the event to the driver again. The driver will detect that it is no longer connected to the Google domain and establish a fresh connection.

The driver will report invalid xml conditions such as invalid class names, attribute names or values with an Error status level.

All other errors will be reported with a Java exception or a Google API exception along with the Status Level and Status Type.

7.2 Java Exceptions

Exception	Cause	Status Level
Java.io.IOException	Interrupted I/O operations	Retry

7.3 Google Directory API Exceptions

The Directory API communicates the result of API operations using either *GoogleJsonResponseException* or *HttpResponseException* objects

HTTP Status-Code	Cause	Status Level
400: Bad Request	The Google server was unable to recognize and process the request.	Error
401: Unauthorized	The Client ID provided by the driver is not authorized to access the Google resource specified in the request. ReviewSection 3.2, “Configuring OAuth2 authentication for Google APIs,” on page 20	Error
403: Forbidden	Error 403: Forbidden can occur from three different situations in the driver: ◆ The Client ID provided by the driver is not authorized to access the Google resource specified in the request. ReviewSection 3.2, “Configuring OAuth2 authentication for Google APIs,” on page 20 with specific emphasis on Section 3.2.2, “Delegate Domain-wide Administrative rights to the Google Service Account,” on page 30 ◆ A query was issued against a domain that doesn’t exist. This includes domaina Google domain alias. ◆ The request has exceeded a Google API Rate limit. The driver will automatically implement Google’s exponential backoff algorithm while retrying the request. If after 5 internal retries the request continues to fail, the driver will return a Retry to the engine.	◆ Access forbidden: Retry ◆ Querying invalid domain: Success. ◆ Google Rate or Limits Exceeded: Retry

HTTP Status-Code	Cause	Status Level
401: Not Found	The Google Server was unable to retrieve the requested resource.	Error
405: Bad Method	The Google Server does not support the HTTP method called.	Error
406: Not Acceptable	The Google Server determined the response type is not supported by the driver.	Error
407: Proxy Authentication Required	Access to the resource requires proxy authentication.	Error
408: Request Timeout	The Google Server has ‘timed-out’ on the socket.	Retry
409: Conflict	The Google Server has been asked to add an object that already exists. ◆ If the exception occurs adding a member to a group, this results in a Warning status. ◆ If the exception occurs adding a User, Group or Organization Unit, this results in an Error status.	◆ If the exception occurs adding a member to a group, this results in a Warning status. ◆ If the exception occurs adding a User, Group or Organization Unit, this results in an Error status.
410: Gone	The Google Server is unable to retrieve the requested object. This is similar to a 401 error.	Error
411: Length Required	The request sent to the Google Server should contain a ‘Content-Length’ attribute.	Error
412: Precondition Failed	The Google Server has a precondition specified on the request was not met.	Error
413: Entity Too Large	The request sent to the Google Server was too large in bytes.	Error
414: Request URI Too Long	The URI requested was too long in bytes.	Error
415: Unsupported Type	The Google Server determined the media type of the object is unsupported.	Error
503: Unavailable	The Google Server is unavailable.	Retry

7.4 Google GData Exceptions

Exception	Cause	Status Level
Java.io.IOException	Interrupted I/O operations	Retry

Exception	Cause	Status Level
com.google.gdata.util.ServiceException	An error occurred in Google while processing a GData request to the Contacts API	Error
com.google.gdata.util.AuthenticationException	This is a connection exception received from Google after the driver has successfully authenticated.	Retry
com.google.gdata.util.InvalidEntryException	The Google Contact Entry ID requested is invalid	Error
com.google.gdata.util.ResourceNotFoundException	This exception indicates that a query failed to retrieve a valid Contact object	If the exception is a result of a query the status level is Success, since a query that doesn't resolve to an object is not an error. If the exception is a result of requesting a Google object based on an Association value the Status Level will be Error.
com.google.gdata.util.ServiceException with an error description of "Internal Server Error"	The Google GData APIs encountered an undefined server error when processing a request.	Retry
Java.net.MalformedURLException	Indicates a malformed URL was received.	Error
com.google.gdata.data.AppsForYourDomainException	An exception thrown by AppsForYourDomainService. This can occur when the driver is operating on Contact objects. .	The Status Level is dependent on the error code associated with the exception.
Unknown Error	The Contacts API is reporting an unknown error condition. This is routinely a transient condition.	Retry
Entity does not exist	An exception occurred looking up or querying for an object.	Success if the operation was a query operation. Error if the operation was a lookup based on an association value.
Entity Exists	An attempt to create a Contact in Google has failed because an object of that name already exists.	Error

7.5 Common Driver Issues

Issue	Example
User Placement. Do not use a leading "\" to place users or Organization Units.	To place a user in the root container, the dest-dn should only contain the Username. If you are placing a user in the google Sales\Marketing container your dest-dn should look like: <add class-name="User" dest-dn="Sales\Marketing\ddare"/> Organization Units use the same format for dest-dn.
Group Placement: Do not use a placement rule on groups as Google does not support placing groups in organizations.	
Group renames are not supported.	The naming attribute of a group in Google is the email address. Google does not support changing this address after the group has been created. It is up to the deveoper as to how best to capture this event. If the group in eDirectory is renamed the driver will continue to manage the group in Google, but the Google group won't be renamed.
Unique naming: It is important that Nicknames, Group names and usernames be unique in the Google apps domain.	When developing a matching rule be sure to check for nicknames and usernames to ensure proper matching. Further, naming must be unique across all Google Organization units. It is not legal to have Sales\Marketing\ddare and Engineering\ddare since ddare needs to be unique across the domain.

7.6 Troubleshooting Driver Processes

Viewing driver processes is necessary to analyze unexpected behavior. To view the driver processing events, use DSTrace. You should only use it during testing and troubleshooting the driver. Running DSTrace while the drivers are in production increases the utilization on the Identity Manager server and can cause events to process very slowly. For more information, see "Viewing Identity Manager Processes" in the *NetIQ Identity Manager 4.0.2 Common Driver Administration Guide*.

A Driver Properties

This section provides information about the Driver Configuration and Global Configuration Values properties for the Google Apps driver. These are the only unique properties for drivers. All other driver properties (Named Password, Engine Control Values, Log Level, and so forth) are common to all drivers. Refer to "Driver Properties" in the *NetIQ Identity Manager 4.0.2 Common Driver Administration Guide* for information about the common properties.

The information is presented from the viewpoint of iManager. If a field is different in Designer, it is marked with an icon.

A.1 Driver Configuration

In iManager:

1. Click to display the Identity Manager Administration page.
2. Open the driver set that contains the driver whose properties you want to edit:
   - **2a** In the **Administration** list, click **Identity Manager Overview**.
   - **2b** If the driver set is not listed on the **Driver Sets** tab, use the **Search In** field to search for and display the driver set.
   - **2c** Click the driver set to open the Driver Set Overview page.
3. Locate the driver icon, then click the upper right corner of the driver icon to display the **Actions** menu.
4. Click **Edit Properties** to display the driver's properties page.

   By default, the Driver Configuration page is displayed.

In Designer:

1. Open a project in the Modeler.
2. Right-click the driver icon or line, then select click **Properties > Driver Configuration.**

The Driver Configuration options are divided into the following sections:

A.1.1 Driver Module

The driver module changes the driver from running locally to running remotely or the reverse.

Table A-1 Driver Module

Option	Description
Java	Used to specify the name of the Java class that is instantiated for the shim component of the driver. This class can be located in the `classes` directory as a class file, or in the `lib` directory as a `.jar` file. If this option is selected, the driver is running locally. The Java class name is: `com.novell.nds.dirxml.driver.gmailshim.GMailDriverShim`
Native	This option is not used with the Google Apps driver.
Connect to Remote Loader	Used when the driver is connecting remotely to the connected system. Designer includes two suboptions: • Driver Object Password: Specifies a password for the Driver object. If you are using the Remote Loader, you must enter a password on this page. Otherwise, the remote driver does not run. The Remote Loader uses this password to authenticate itself to the remote driver shim. • Remote Loader Client Configuration for Documentation: Includes information on the Remote Loader client configuration when Designer generates documentation for the driver.

A.1.2 Driver Object Password

Table A-2 Driver Object Password

Option	Description
Driver Object Password	Use this option to set a password for the driver object. If you are using the Remote Loader, you must enter a password on this page or the remote driver does not run. This password is used by the Remote Loader to authenticate itself to the remote driver shim.

A.1.3 Authentication

The authentication section stores the information required to authenticate to the connected system.

Table A-3 Authentication

Option	Description
Authentication ID or User ID	This is a User ID on the target Google domain that has administrative rights on the domain. The driver will authenticate to Google Apps using this User ID. If your domain is mydomain.com, then this user id would be in the form: admin@mydomain.com

Option	Description
Authentication Context or Connection Information	This is the name of the Google domain to be managed by the driver. If your Google domain is named mydomain.com, then you would enter mydomain.com in the Authentication Context.
Remote Loader Connection Parameters or Host name Port KMO Other parameters	Used only if the driver is connecting to the application through the remote loader. The parameter to enter is `hostname=xxx.xxx.xxx.xxx port=xxxx kmo=certificatename`, when the host name is the IP address of the application server running the Remote Loader server and the port is the port the remote loader is listening on. The default port for the Remote Loader is 8090. The `kmo` entry is optional. It is only used when there is an SSL connection between the Remote Loader and the Metadirectory engine. Example: `hostname=10.0.0.1 port=8090 kmo=IDMCertificate`
Driver Cache Limit (kilobytes) or Cache limit (KB)	Specify the maximum event cache file size (in KB). If it is set to zero, the file size is unlimited. Click Unlimited to set the file size to unlimited in Designer.
Application Password or Set Password	This option is not used with the Google Apps driver. Application authentication is accomplished using OAuth.
Remote Loader Password or Set Password	Used only if the driver is connecting to the application through the Remote Loader. The password is used to control access to the Remote Loader instance. It must be the same password specified during the configuration of the Remote Loader on the connected system.

A.1.4 Startup Option

The Startup Option section allows you to set the driver state when the Identity Manager server is started.

Table A-4 *Startup Option*

Option	Description
Auto start	The driver starts every time the Identity Manager server is started.
Manual	The driver does not start when the Identity Manager server is started. The driver must be started through Designer or iManager.
Disabled	The driver has a cache file that stores all of the events. When the driver is set to Disabled, this file is deleted and no new events are stored in the file until the driver state is changed to Manual or Auto Start.
Do not automatically synchronize the driver	This option only applies if the driver is deployed and was previously disabled. If this is not selected, the driver re-synchronizes the next time it is started.

A.1.5 Driver Parameters

Table A-5 Parameter Name

Parameter Name	Description
Service Account Email Address	Set this parameter to the service account email address created for your service account credential. See Section 3.2.1, “Creating a Google Service Account,” on page 20
P12 Private Key File	Set this parameter to the path and filename of the .p12 file created for your service account credential. Section 3.2.1, “Creating a Google Service Account,” on page 20
Override JAXP Parameter	Set to `true` to have the driver override the default setting for the system property **javax.xml.parsers.SAXParserFactory** with the value `org.apache.xerces.jaxp.SAXParserFactoryImpl`.
Hash Passwords	Setting this subscriber parameter to True tells the driver to apply an MD5 hash to the password before passing it to Google.
Heartbeat Interval	This publisher parameter tells the publisher how frequently to emit a heartbeat document to the IDM engine.

A.2 Global Configuration Values

Global configuration values (GCVs) are values that can be used by the driver to control functionality. GCVs are defined on the driver or on the driver set. Driver set GCVs can be used by all drivers in the driver set. Driver GCVs can be used only by the driver on which they are defined.

The Google Apps driver includes several predefined GCVs. You can also add your own if you discover you need additional ones as you implement policies in the driver.

To access the driver’s GCVs in iManager:

1. Click  to display the Identity Manager Administration page.
2. Open the driver set that contains the driver whose properties you want to edit.
   - **2a** In the **Administration** list, click **Identity Manager Overview**.
   - **2b** If the driver set is not listed on the **Driver Sets** tab, use the **Search In** field to search for and display the driver set.
   - **2c** Click the driver set to open the Driver Set Overview page.
3. Locate the driver icon, click the upper right corner of the driver icon to display the **Actions** menu, then click **Edit Properties**.

   or

   To add a GCV to the driver set, click **Driver Set**, then click **Edit Driver Set properties**.

To access the driver’s GCVs in Designer:

1. Open a project in the Modeler.

2 Right-click the driver icon or line, then select **Properties > Global Configuration Values.**

or

To add a GCV to the driver set, right-click the driver set icon, then click **Properties > GCVs**.

Table A-6 *Global Configuration Values*

Name	Description	Example Value
Google Apps Domain Name	Specify the name of the Google Apps domain managed by this driver.	mydomain.com
Google Apps Secondary Domain Name	Multi-valued list of secondary Google Apps Domain names associated with the primary domain.	mysubdomain.com
How should invalid username characters be handled?	Google only allows the following characters in usernames: a-z, A-Z, 0-9, hyphen - , underscore _, apostrophe ‘, and period . Periods may not be sequential (. is ok, .. is not allowed). The connector will test users during creation for valid characters in their username. Choose the desired method of handling invalid characters: ◆ **Strip invalid characters and create user.** This method will remove invalid characters from the username with no substitution. ◆ **Block creation of users with invalid usernames.** When an invalid character is detected, the user creation will be vetoed (blocked) by the connector. It will not be created in Google until the username is valid and the object is resent to the connector.	**Strip invalid characters and create user.** **Block creation of users with invalid usernames.**
Use Entitlements to Control GoogleApps Accounts?	If this GCV is set to true then users will only be created in Google when the entitlement is granted.	**True** or **False**
Match Users who do not have a Google Account Entitlement.	If this GCV is set to true then users who have not been given an entitlement will be matched to existing Google Accounts. Otherwise the connector will not attempt to match users without a Google Account Entitlement they will just be blocked at the matching rule.	False

Name	Description	Example Value
What should the Connector do when the Google Account entitlement is revoked?	This GCV determines how the connector will handle a user account who has their Account Entitlement revoked. Do Nothing: This means that if an Account Entitlement is revoked, then the driver will do nothing. The account will remain in the state it was in when it was revoked. Disable Account: If this is selected then the entitlement is revoked. The account in Google will be disabled.Delete Account: This will tell the connector to Delete the account in Google when the entitlement is revoked.	Do Nothing
Use Group Membership Entitlement	Users will only be added to group membership if this entitlement is set to **true**	false
Base Container for users in eDirectory	Only users in or below this container will be synchronized to the connected Google System.	yourorg\users
Base Container for Groups in eDirectory	Only Groups in or below this container will be synchronized to the connected Google System.	Yourorg/groups
Base Container for Organizational Units in eDirectory	Only OU's in or below this container will be synchronized to the connected Google System. If placement is done with mirroring package this GCV is also used as the root container for where the mirror will start.	Myorg
What to use for intitial Password if Distribution Password not Present	If the system is not set up for universal password synchronization or the user account just doesnt have a distribution password set yet, then an initial password has to be set. This GCV tells the system whether to use an attribute off of the user account for an initial password or to use a random generated password. If the accounts are going to use SAML for authentication then a Random Password would be fine. Otherwise an attribute value should be selected.	Random Password Attribute Value from User
eDirectory attribute to use for initial password value.	This is the name of the attribute in edirectory that the Google driver should use for an initial password if no Distribution password is available on creation.	Surname
Number of letters to use in the Random Password	This is the number of Letters to use in the random password. when added to the value of the "Random password numbers" GCV It will determine the number of characters in the total Length	6

Name	Description	Example Value
Number of numbers to use in the Random Password	This is the number of numbers to use in the random password. when added to the value of the "Random password letters" GCV It will determine the number of characters in the total Length.	6

A.3 Special Attributes

The driver exposes attributes that either do not map directly to eDirectory attributes or have special handling in the driver.

A.3.1 ExternalId

ExternalId is a structured, multi-valued attribute which the driver maps to the UserExternalId object in the Google Directory API. An ExternalId references an identifier outside the Google domain. An ExternalId is made up of an ID type and a value.

Valid types are:

- account
- custom
- customer
- network
- organization

ExternalId is sent to the driver as a structured type.

```
<modify-attr attr-name="ExternalId">
      <add-value>
          <value timestamp="1467727743#2" type="structured">
               <component name="value">bob@dog.com</component>
               <component name="type">account</component>
           </value>
      </add-value>
  </modify-attr>
```

If the type is `custom` then a 3rd component with `name=customtype` containing the name of the custom type must be provided.

A.3.2 WorkFormattedAddress and HomeFormattedAddress

The address value displayed in the Google Admin U/I is actually the `Formatted` field of a `UserAddress` object. The U/I will show a UserAddress object of type=work and type=home, as well as custom types.

The driver generates a value for the Formatted field when it detects a change to one of the fields of a UserAddress object. The generated value is simply a concatenation, without delimiters, of the fields present on a UserAddress object. We recognize that this format may not meet every organization's needs. So we have also added `WorkFormattedAddress` and `HomeFormattedAddress` in the driver's schema. This allows an organization to directly map an attribute to update WorkFormattedAddress or HomeFormattedAddress with an address formatted to meet their needs.

A.3.3 GMail Settings API Attributes

Several attributes are exposed for the Google Schema that update a users default email settings within a Google Domain. These attributes are not mapped to an eDirectory attribute but can be sent on modify or add events. These attributes are:

- GmailSettingsEnableIMAP
- GmailSettingsEnablePop
- GmailSettingsForwarding
- GmailSettingsLabel
- GmailSettingsEnableLanguage
- GmailSettingsSendAs
- GmailSettingsSignature

Table A-7 *Special Attributes*

Setting	Example

GmailSettingsEnableIMAP
 	
Turns on or off IMAP for the Account. Set to True or False.
	`<add-attr attr-name="GmailSettingsEnableIMAP">`
	`  <value type="string">true</value>`
	`</add-attr>`

GmailSettingsEnablePOP
 	
Turns on or off POP for the Account.
	`<add-attr attr-name="GmailSettingsEnablePOP">`
	`  <value type=″structured″>`
	`  <component name="EnableFor">Don DaRe</component>`
	`  <component name="Action">don@idmtest.org</component>`
	`  <component name="Enable">true</component>`
	`  </value>`
	`</add-attr>`

GmailSettingsForwarding
 	
Sets a forwarding email address. Note the API only allows setting this to an account inside of the Google Apps Domain. External addresses will cause an error.
	`<add-attr attr-name="GmailSettingsForwarding">`
	`  <value type=″structured″>`
	`  <component name="ForwardAddress">Don DaRe</component>`
	`  <component name="Action">don@idmtest.org</component>`
	`  <component name="Enable">true</component>`
	`  </value>`
	`</add-attr>`

Setting	Example
GmailSettingsLabel This is a set of labels that will be automatically set on the account. The labels will be available in gmail to the end user. .	<add-attr attr-name="GmailSettingsLabel"> <value type=”string”MyProject</value> </add-attr>
GmailSettingsLanguage This sets the default language for the user.	<add-attr attr-name="GmailSettingsLanguage"> <value type=”string”Eng</value> </add-attr>
GmailSettingsSendAs Set this structured value to setup a send as alias. Useful when there are multiple domains or subdomains in Google Apps..	<add-attr attr-name="GmailSettingsSendAs"> <value type=”structured”> <component name="Name">Don DaRe</component> <component name=”SendAs”>don@idmtest.org</ component> <component name=”ReplyTo”>Don@idmtest.org</ component> <component name=”IsDefault”>true</component> </value> </add-attr>
GmailSettingsSignature Set a default email signature on the user. This is at the user level and can be overridden by the end user.	<add-attr attr-name="GmailSettingsSignature"> <value type=”string”>Signature Data</value> </add-attr>