

Users



gam **create** | **update** user <user email> [options]
 gam <who> **update** user [options]

Common options:

[firstname <First Name>] [lastname <Last Name>]
 [password <Password>] [email <New Email>]
 [gal on|off] [admin on|off] [suspended on|off]
 [sha] [md5] [crypt] [nohash]
 [changepassword on|off] [org <Org Name>]

Extended options:

[relation <relation type> <relation value>]
 [externalid <id type> <id value>]
 [phone type <phone type> value <phone value>
 primary|notprimary]
 [organization name <org name> title <org title>
 type <org type> department <org dept>
 symbol <org symbol> costcenter <org cost center>
 location <org location>description <org desc>
 domain <org domain> primary|notprimary]
 [address type <address type>
 unstructured <unstructured address>
 extendedaddress <address extended address>
 streetaddress <address street address>
 locality <address locality> region <address region>
 postalcode <address postal code> pobox <address pobox>
 countrycode <address country code> primary|notprimary]
 [im type <im type> protocol <im protocol>
 primary <im value>]
 [location type <location type> area <area>
 building <building> desk <desk> floor <floor>
 section <section> endlocation]
 [sshkeys expires <date> key <keyvalue>]
 [posixaccounts geccos <geccos> gid <numeric gid>
 uid <numeric uid> home <home path>
 primary true|false shell <shell> system <systemid>
 username <username> endposix]
 [agreedtoterms on|off]
 [schemaname.fieldname <fieldvalue>]
 [schemaname.multivaluefieldname multivalued
 <fieldvalue>]

Extended options for update only:

[customerid <string>]
 [otheremail home|work|other|<custom> <email address>]

gam **info** user <user email> [nogroups] [noaliases]
 [nolicensess] [noschemas] [schemas list,of,schemas]
 [userview] [skus <list,of,skus>]

gam **delete** user <user email>

gam **undelete** user <user email> [org <org Name>]

gam <who> **print**

gam **print** users [allfields] [custom all|list,of,schemas]
 [userview] [ims] [emails] [externalids] [relations]
 [addresses] [organizations] [phones] [licenses] [photo]
 [firstname] [lastname] [emailparts] [deleted_only] [id]
 [orderby email|firstname|lastname] [query <query>]
 [ascending|descending] [domain <Domain Name>]
 [fullname] [ou] [suspended] [changepassword] [gal]
 [agreed2terms] [admin] [creationtime] [aliases]
 [lastlogintime] [groups] [ismailboxsetup] [todrive]

gam <who> **show** gmailprofile | gplusprofile [todrive]

User Email

gam <who> <attributes/values>

where attributes and values are:

language <language code>

pagesize 25|50|100

shortcuts | **arrows** | **snippets** | **utf** | **webclips** on|off

signature <signature text> | (file <signature file>

[charset <Charset>]) (replace <Tag> <String>)*

[name <String>] [replyto <EmailAddress>]

vacation on|off [subject <String> (message <String>)|

(file <FileName> [charset <CharSet>]) (replace <Tag>

<String>)* [html] [contactsonly] [domainonly]

[startdate <Date>] [enddate <Date>]]

[add **label** <label name> [messagelistvisibility hide|show]

[labellistvisibility hide|show|showifunread]

delete label <label name> | regex:<RegularExpression> |

--ALL_LABELS--

update label search <search> replace <replace> [merge]

update labelsettings <label name> name <new label name>

[message_list_visibility show|hide]

[label_list_visibility show|hide|show_if_unread]

show labels [onlyuser]

filter from <email address> | to <email address> |

subject <words> | haswords <words> |

nowords <words> | musthaveattachment |

label <label name> | markread | archive | star |

forward <email address> | trash | neverspam

delete | info **filter** <FilterIDEntity>

[add] | update **sendas** <email address> <name> [default]

[replyto <email address>] [treatasalias <Boolean>]

[signature <String>] (file <FileName> [charset <CharSet>])

(replace <RegularExpression> <String>)*]

delete sendas <email address>

info **sendas** <email address> [format]

pop on|off [for allmail|newmail|fromnowown] [action

keep|leaveinbox|archive|delete|trash|markread]

imap on|off [noautoexpunge]

[expungebehavior archive|deleteforever|trash]

[maxfoldersize 0|1000|2000|5000|10000]

forward on|off [email address] [keep|archive|delete]

delegate to <user email>

add **delegate** | **forwardingaddress** <user email>

show **delegates** [csv]

show **forwardingaddress**

show **vacation** | **filters** | **imap** | **pop** | **forward** | **profile**

show **sendas** | **vacation** | **signature** [format]

print **delegates** | **forwardingaddress** [todrive]

delete delegate | **forwardingaddress** <user email>

profile shared|unshared

update **photo** <photo filename>

get **photo** [drivedir | (targetfolder <FilePath>)] [noshow]

delete photo

print **filters** | **forward** | **sendas** [todrive]

gam user <who> **delete** | **trash** | **untrash** messages | threads

query <gmail search> [doit] [maxto<action> <number>]

gam user <who> **modify** messages

addlabel | removelabel <label> query <gmail search>

[doit] [maxto<action> <number>]

where <action> is:

delete | trash | untrash | modify

e.g. maxtotrash, maxtountrash

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.

Calendars

31

gam calendar <calendar email> <action> [<options>]

where action and options are:

showacl | **wipe**

add | **update** freebusy | read | editor | owner <user email> |
user <user email> | group <group email> |
domain [<Domain name>] | default

delete <user email> | user <user email> |
group <group email> | domain [<Domain name>] | default

addevent [attendee <user email>] [location <location>]
[optionalattendee <user email>] [anyonecanaddself]
[summary <summary>] [source <title> <url>]
[description <event description>]
[id <id>] [available]

start allday <YYYY-MM-DD> | <start datetime>
end allday <YYYY-MM-DD> | <end datetime>
[guestscantinviteothers] [guestscantseeothers]
[visibility default | public | private] [tentative]
[notifyattendees]
[noreminders] |
[reminder <minutes> email | popup | sms]
[recurrence <repeat>] [timezone <timezone>]
[privateproperty <Key> <Value>]
[sharedproperty <Key> <Value>]
[colorindex <Number from 1 to 11>]

deleteevent [eventid <id>] [query <query>]
[notifyattendees] [doit]

gam <who> **show** calendars | calsettings

gam <who> **delete** calendar <calendar email>

gam <who> **add** | **update** calendar <calendar email>
[selected true | false] [hidden true | false]
[reminder email | sms | popup (minutes)]
[summary <summary>] [colorindex (1-24)]
[backgroundcolor <htmlcolor>
[foregroundcolor <htmlcolor>]

gam <who> **update** calattendees csv <csv file>
[start YYYY-MM-DD] [end YYYY-MM-DD]
[allevents] [dryrun]

gam <who> **transfer** seccals <target user> [keepuser]

gam <who> **info** calendar <calendar email> | primary

gam <who> **print** calendars [todrive]

Drive Management

gam <who> **show** filelist [todrive] [query <query>]
[allfields] [createddate] [description] [fileextension]
[filesize] [id] [restricted] [starred] [trashed] [viewed]
[lastmodifyingusername] [lastviewedbydate]
[modifieddate] [originalfilename] [quotaused] [shared]
[writerscanshare]

gam <who> **show** driveactivity [todrive] [fileid <id>]
[folderid <id>]

gam <who> **show** drivesettings [todrive]

gam <who> **show** fileinfo <id> |
[allfields | <DriveFieldName>*]

gam <who> **show** filetree

gam <who> **show** filerevisions <id>

gam <who> **add** drivefile
[localfile <filepath>] [drivefilename <filename>]
[convert] [ocr] [ocrlanguage <language>]
[restricted] [starred] [trashed] [viewed]
[lastviewedby <date>] [modifieddate <date>]
[description <description>] [mimetype <type>]
[parentid <folder id>] [parentname <folder name>]
[writerscantshare]

gam <who> **update** drivefile
[localfile <filepath>] [newfilename <filename>]
[id <drive file id> | drivefilename <filename>]
[convert] [ocr] [ocrlanguage <language>]
[restricted true | false] [starred true | false]
[trashed true | false] [viewed true | false]
[lastviewedby <date>] [modifieddate <date>]
[description <description>] [mimetype <type>]
[parentid <folder id>] [parentname <folder name>]
[writerscantshare]

gam <who> **get** drivefile [id <file id> | query <query>]
[format <openoffice | microsoft | pdf>]
[targetfolder <local path>] [revision <number>]

gam <who> **delete** emptydrivefolders |
drivefile <file id> [purge]

gam <who> **transfer** drive <target user> [keepuser]

gam <who> **empty** drivetrash

ACLs

gam user <user email> **show** drivefileacl <file id> [asadmin]

gam user <user email> **add** drivefileacl <file id>
[user | group | domain <value> | anyone] [withlink]
[role <reader | commenter | writer | owner>] [sendemail]
[emailmessage <message text>]

gam user <user email> **update** drivefileacl <file id>
<permission id> [withlink]
[role <reader | commenter | writer | owner>]
[transferownership <true | false>] [asadmin]

gam user <user email> **delete** drivefileacl <file id>
<permission id> [asadmin]

Team Drive

gam user <email> **add** teamdrive <name>

gam user <email> **update** teamdrive <id> [name <name>]

gam user <email> **delete** teamdrive <id>

gam user <email> **print** | **show** teamdrives
[todrive] [asadmin]

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.

Groups



```
gam create group <group email> <GroupAttributes>
gam update group <group email>
  [admincreated true|false] [email <EmailAddress>]
  <GroupAttributes>
gam info group <group email> [nusers] [noaliases]
  [groups]
gam update group <group email>
add | update | sync
owner|member|manager
{user <email address> |
  group <group address> | org <org name> |
  file <file name> | all users}
gam update group <group email> clear
  [owner] [manager] [member]
gam update group <group email>
remove {user <email address> |
  group <group address> | org <org name> |
  file <file name> | all users}
gam delete group <group email>
gam <who> delete group
gam print groups [domain <domain>]
  [member <user email>] [maxresults <results>]
  [name] [description] [admincreated] [id] [aliases]
  [members] [owners] [managers] [settings] [todrive]
  [delimiter <delimchar>] [fields <list,of,fields>]
gam print group-members [group <group email>]
  [domain <domain>] [member <user email>]
  [fields <list,of,fields>] [membenames] [todrive]
```

Calendar Resources

```
gam create resource <id> <Name> [parameters]
gam update resource <id> [name <Name>] [parameters]
where parameters are:
[description <description>] [type <type>]
[building <building>] [capacity <number>]
[features <features>] [floor <floor>]
[floorsection <floorsection>]
gam info | delete resource <id>
gam print resources [id] [description] [email] [type]
  [allfields] [todrive]
gam create building <name> [id <id>] [latitude <latitude>]
  [longitude <longitude>] [description <description>]
  [floors <floors>]
gam update building <name or id:id> [name <name>]
  [latitude <latitude>] [longitude <longitude>]
  [description <description>] [floors <floors>]
gam info building <name or id:id>
gam delete building <name or id:id>
gam create feature name <name>
gam update feature <name> name <newname>
gam delete feature <name>
```

Group Settings

```
gam create | update group <group email>
  <attributes/values>
```

where attributes and values are:

```
allow_external_members true|false
message_moderation_level moderate_all_messages|
  moderate_new_members|moderate_none|
  moderate_non_members
primary_language <language>
reply_to reply_to_custom|reply_to_ignore|
  reply_to_list|reply_to_managers|reply_to_owner|
  reply_to_sender
send_message_deny_notification true|false
show_in_group_directory true|false
who_can_invite all_managers_can_invite|
  all_members_can_invite
who_can_join all_in_domain_can_join|
  anyone_can_join|can_request_to_join|
  invited_can_join
who_can_post_message all_in_domain_can_post|
  all_managers_can_post|all_members_can_post|
  anyone_can_post|none_can_post
who_can_view_group all_in_domain_can_view|
  all_managers_can_view|all_members_can_view|
  anyone_can_view
who_can_view_membership all_in_domain_can_view|
  all_managers_can_view|all_members_can_view
allow_google_communication true|false
allow_web_posting true|false
archive_only true|false
custom_reply_to <email address>
is_archived true|false
max_message_bytes <integer>
members_can_post_as_the_group true|false
message_display_font default_font|fixed_width_font
description <group description>
name <new name>
spam_moderation_level allow|moderate|
  silently_moderate|reject
include_in_global_address_list true|false
who_can_leave_group none_can_leave|
  all_members_can_leave|all_managers_can_leave
who_can_contact_owner anyone_can_contact|
  all_in_domain_can_contact|
  all_members_can_contact|all_managers_can_contact
```

Email Aliases

```
gam create | update alias <alias> user | group | target
  <primary address>
gam info alias <alias>
gam delete alias [user | group | target] <alias>
gam <who> delete alias
gam print aliases [todrive]
```

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.

Classroom



gam **create** course [id|alias <alias>] <CourseAttributes>
gam **update** course <CourseID> | alias <CourseAttributes>
where <CourseAttributes> is:
[name <name>] [section <section>]
[teacher <teacher email>]
[heading <heading>] [description <description>]
[room <room>] [status <PROVISIONED|ACTIVE|
ARCHIVED|DECLINED>]
gam **info** | **delete** course <CourseID>
gam course <CourseID> <command>
where <command> is:
add student|teacher <email address>
add | **delete** alias <alias>
sync students|teachers group <group email> |
ou <orgunit> | file <filename> |
query <users query> | course <id or alias>
remove student|teacher <email address>
gam **print** courses [teacher <email>] [student <email>]
[state <states>] [todrive] [aliases] [delimiter <String>]
gam **print** course-participants [course <id or alias>]
[student <email>] [teacher <email>]
[show all|students|teachers] [todrive]
gam **create** | **delete** guardian <guardian email>
<student email>
gam **show** guardian [invitedguardian <guardian email>]
[student <StudentItem>] [invitations]
[states <GuardianStateList>] [<UserTypeEntity>]
gam **print** guardians [invitations] [student <email>]
[invitedguardian <email>]
[user <username> | group <email> |
ou <ouname> | all users] [nocsv] [todrive]
[states <GuardianStateList>]
where GuardianStateList is:
COMPLETE | PENDING |
GUARDIAN_INVITATION_STATE_UNSPECIFIED
gam **cancel** guardianinvitation | guardianinvitations
<GuardianInvitationID> <student email>

Data Transfers

gam **create** datatransfer <old owner> drive | google+ |
calendar <new owner> [parameter <value>]
gam **info** datatransfer <id>
gam **print** datatransfers [oldowner <old owner>]
[newowner <new owner>] [status completed|failed|
InProgress|pending] [todrive]
gam **print** transferapps

User Security Settings

gam <who> | file <filename> <command>
where <command> is:
show tokens (oauth tokens)
show token clientid <client id>
delete token clientid <client id>
show asps (app-specific pwds)
delete asp <ID>
show backupcodes (2-step verification)
update backupcodes
delete backupcodes
deprovision (remove tokens, asps,
and backupcodes)
gam **print** tokens [todrive] [clientid <ClientID>] <who>
gam <who> **show** tokens
S/MIME
gam user <email> **add** smime <file <filename>>
<password <password>> [default] [sendas <email>]
gam user <email> **update** smime [id <id>]
[sendas <email>] <default>
gam user <email> **delete** smime <id <id>>
[sendas <email>]
gam user <email> **show** | **print** smime primaryonly todrive

Reports

gam **report** users [todrive] [date <YYYY-MM-DD>]
[user <user email>] [filter <filter terms>]
[fields <included fields>]
gam **report** admin | calendar | logins | tokens | drive |
groups | mobile [todrive] [user <user email>]
[filter <filter terms>] [ip <ip address>]
[event <event name>]
[start <start datetime>] [end <end datetime>]
where the filter specifies the search terms¹.
gam **report** domain [todrive] [date <YYYY-MM-DD>]
[fields <included fields>]
<start datetime> and <end datetime> are in the format:
YYYY-MM-DDThh:mm:ss.000Z

¹ <https://developers.google.com/admin-sdk/reports/v1/reference/userUsageReport/get>

Admin Roles

gam **create** admin <user> <role>
<customer | org_unit <OU>>
gam **delete** admin <role assignment id>
gam **print** admins [user <user>] [role <role>] [todrive]
gam **print** roles [todrive]

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.

Chrome Devices

gam update cros <device id> [user <user info>]
[location <location info>] [notes <notes info>]
[ou <new org unit>] [assetid <asset id>]
[status active | deprovisioned | inactive | returnapproved |
returnrequested | shipped | unknown]

gam update cros <device id>
action disable | reenable |
deprovision_same_model_replace |
deprovision_different_model_replace |
deprovision_retiring_device
[acknowledge_device_touch_requirement]

gam info cros <device id>
[allfields | full | basic] [nolists] [listlimit <Number>]
<CrOSFieldName>* [fields <CrOSFieldNameList>]

gam print cros [query <query>]
[orderby location | user | lastsync | serialnumber |
supportenddate]
[ascending | descending] [todrive]
[allfields | full | basic] [nolists]
[listlimit <Number>] [limit_to_ou <OU>]
<CrOSFieldName>*
[fields <CrOSFieldNameList>]

gam print crosactivity [query <query>] [todrive] [times]
[users] [start <yyyy-mm-dd>] [end <yyyy-mm-dd>]

Mobile Devices

gam update mobile <mobile id>
action **wipe** | **approve** | **block** |
cancel_remote_wipe_then_activate |
cancel_remote_wipe_then_block
[model <String>] [os <String>] [useragent <String>]

gam info mobile <mobile id>

gam print mobile [query <query>][basic | full]
[orderby deviceid | email | lastsync |
model | name | os | status | type]
[ascending | descending] [todrive]

gam delete mobile <mobile id>

Organizational Units

gam create org <name> [description <Description>]
[parent <Parent Org>] [inherit | noinherit]

gam update org <name> [name <Name>]
[description <Description>] [parent <Parent>]
[inherit | noinherit]

gam update org <name> **add** | **move** users <Users> |
file <File Name> | group <Group Name> |
all cros | cros <device id>

gam info org <name> [nousers | child]

gam delete org <name>

gam print orgs [todrive] [toplevelonly]
[fromparent <name>] [allfields | <list of fields>]

CloudPrint

gam printer register

gam update printer <id> [quotaEnabled <true | false>]
[dailyQuota <number>] [public <true | false>]
[name <name>] [proxy <proxy>]
[description <description>] [currentquota <Number>]
[defaultdisplayname <String>] [displayname <String>]
[firmware <String>] [gcpversion <String>]
[istosaccepted <Boolean>] [manufacturer <String>]
[model <String>] [ownerid <EmailAddress>]
[status <Number>] |[type <String>] |[uuid <String>] |
[setupurl <URL>] |[supporturl <URL>] |
[updateurl <URL>]

gam info printer <id> [everything]

gam delete printer <id>

gam print printers [query <query>] [type <type>]
[status <status>] [extrafields <connectionStatus |
semanticState | uiState | queuedJobsCount>] [todrive]

gam printer <id> <command>

where <command> is:
add USER | MANAGER <group or user email> |
<domain> | public
remove <group or user email> | <domain> | public
showacl

gam printjob <printer id> **submit** <file or url>
[title <string>] [tag <string>]

gam printjob <id> **cancel** | **delete**

gam print printjobs [older_than <number><m | h | d>]
[newer_than <number><m | h | d>] [query <query>]
[status <status>] [printer <printer id>]
[owner <user email>] [limit <number>]
[orderby create_time | status | title]
[ascending | descending] [todrive]

gam printjob <PrintJobID> **resubmit** <PrinterID>

gam printjob <PrinterID> | any **fetch**
[olderthan | newerthan <number><m | h | d>]
[query <query>] [status done | error | held | in_progress |
queued | submitted] [orderby create_time | status | title]
[ascending | descending] [owner <user email>]
[limit <number>]

Licenses

gam <who> **add** | **delete** license <sku>

gam <who> **update** license <sku> [from] <oldsku>

gam print licenses [products <product>] [sku <sku>]
[todrive]

where <sku> is:
free | gafw | gafg | gams | gal | gau | gae | coordinate | vault |
vfe | coordinate | chrome | drive20gb | drive50gb |
drive200gb | drive400gb | drive1tb | drive2tb | drive4tb |
drive8tb | drive16tb

and <product> is
Google-Apps | Google-Drive-storage |
Google-Coordinate | Google-Vault

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.

Domain Settings

gam **info** instance
gam **info** domain | domainalias <domain>
gam **create** domain <domain>
gam **create** domainalias <domainalias> <parentdomain>
gam **delete** domain | domainalias <domain>
gam **print** domains | domainaliases [todrive]
gam **update** domain <domain> primary

Domain Verification

gam **create** verify <domain>
gam **update** verify <domain> <cname | txt | site | file>
gam **info** verify

Admin Notifications

gam **update** notification id all | <notification ID>
unread | read
gam **info** notification [unreadonly]
gam **delete** notification id all | <notification ID>

Reseller

gam **create** | **update** resoldcustomer <domain>
[alternateemail <email>] [phone <phone>]
[address1 <address>] [address2 <address>]
[address3 <address>] [contact <name>]
[country <country>] [city <city>]
[organizationname <name>] [postalcode <code>]
[region <region>] [transfertoken <token>]
gam **info** resoldcustomer <domain>
gam **create** resoldsubscription <domain>
[dealcode <code>] [plan <plan>] [purchaseorderid <po>]
[seats <number> [max]] [sku <sku>]
[transfertoken <token>]
gam **update** resoldsubscription <domain> <sku> [activate]
[suspend] [startpayservice] [renewal <renewaltype>]
[seats <number> [max]] [plan <plan>]
[seats <number> [max]] [purchaseorderid <po>]
[dealcode <code>]]
gam **info** resoldsubscriptions <domain>
[transfertoken <token>]
gam **delete** resoldsubscription <domain> <sku>
<deletiontype>
export CUSTOMER_ID=<customer id>
gam user <email> check serviceaccount
gam user <email> add drivefile localfile <filepath>

Vault

gam create vaultmatter [name <name>]
[description <description>] [collaborators <emails>]
gam update vaultmatter <nameOrId> [name <name>]
[description <description>]
[addcollaborators <emails>]
[removecollaborators <emails>]
[action close | reopen | delete | undelete]
gam info vaultmatter <nameOrId>
gam print vaultmatters [todrive] [basic]
gam create vaulthold <matter nameOrId>
[name <name>] [corpus mail | drive | groups]
[query <query>] [accounts <emails>]
[orgunit <orgunit>]
[starttime YYYY-MM-DD] [endtime YYYY-MM-DD]
gam update hold <HoldNameOrId>
<matter <MatterNameOrId>> [query <query>]
[orgunit <orgunit>] [addaccounts <emails>]
[removeaccounts <emails>]
[starttime YYYY-MM-DD] [endtime YYYY-MM-DD]
gam info | delete vaulthold <HoldNameOrId>
matter <MatterNameOrId>
gam print vaultholds [matters <matters>] [todrive]

Customer

gam **info** customer
gam **update** customer
[alternateemail <EmailAddress>]
[language <LanguageCode>] [phone <String>]
[organizationname <String>] [contactname <String>]
[countrycode <String>] [addressline1 <String>]
[addressline2 <String>] [addressline3 <String>]
[postalcode <String>]

Custom User Schema

gam **create** | **update** schema <schemaname>
field <fieldname>
type <bool | double | email | int64 | phone | string>
[indexed] [restricted] [multivalued]
[range <min number> <max number>] endfield
gam **print** schemas
gam **info** schema <schemaname>
gam **delete** schema <schemaname>

Various

gam **version** [simple] [check]
gam **version** | **help**
gam **create** project
gam **whatis** <user email>

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.

Bulk Operations

gam **csv** <csv-filename> gam <regular command>

Arguments starting with ~ will be substituted with the value from the CSV row. For example:

```
gam csv newusers.csv gam create user
~Email firstname ~Firstname lastname ~Lastname
password ~Password gal off changepassword off
org Testing/Students
```

Note that the ~ (tilde) character may need to be escaped in some environments.

Run multiple gam commands from batch file:

```
gam batch <batch-filename>
```

When the command “commit-batch” is found, all previous commands will be completed before execution continues.

Parallel Commands

For both csv and batch commands, GAM will run multiple actions in parallel. By default, GAM starts 5 worker threads and can run 5 commands at a time. You can raise or lower this setting by setting an environment variable called GAM_THREADS.

Going much higher than 20 threads is not recommended or you’re likely to see issues with Google API quotas.

OAuth Key Management

Set the variable OAUTHFILE in your shell to the file containing the OAuth credentials and the variable OAUTHSERVICEFILE to the file containing the service account JSON key. The file must be in the same folder as gam.exe/gam.py or contain a path relative to this folder.

```
gam oauth info [<access token>]
```

```
gam oauth create | revoke
```

Definitions

<who> user <user email> | group <group email> | ou <ouname> | all users

<datetime> RFC3339 formatted date and time, e.g. YYYY-MM-DDThh:mm:ss.000Z

<user email> is a user in the domain

<group email> is a group in the domain

<email address> is any email address

DISCLAIMER: This is a reference sheet for people familiar with GAM and is provided “as-is”. You use it at your own risk! Always test commands before running them in a production environment.