AD Bridge 22.1

Linux Administration Guide

Table of Contents

AD Bridge Linux Administration Guide

AD Bridge joins Linux and Unix computers to Active Directory so that you can centrally manage all your computers from one source, authenticate users with the highly secure Kerberos protocol, control access to resources, and apply group policies to non-Windows computers.

This guide shows system administrators and security administrators how to use BeyondTrust AD Bridge.

Access Command Line Tools

The AD Bridge command line tools are located at **/opt/pbis/bin**. You can access the tools using either an absolute path or a relative path.

You can access help for any command using **--help**.

Example:

```
/opt/pbis/bin/find-group-by-id --help
```

Note: *Some commands use a different syntax to access help. The syntax is provided in the command description.*

Manage AD Bridge Enterprise Services

The Service Manager lets you track and troubleshoot all AD Bridge services with a single command line utility.

./lwsm list

Lists the status of all known services. Run this command with superuser privileges.

Example:

```
/opt/pbis/bin/lwsm list
```

./lwsm restart <service>

Restarts a service, automatically identifying the service's dependencies and restarting them in the correct order. Run this command with superuser privileges.

Example:

```
/opt/pbis/bin/lwsm restart lsass
```

./lwsm refresh <service>

Refreshes a service's configuration, forcing use of a new configuration after a registry setting has been changed. Run this command with superuser privileges.

Example:

```
/opt/pbis/bin/lwsm refresh lsass
```

./lwsm info <service>

Displays information about a service, including any dependencies.

Example:

```
/opt/pbis/bin/lwsm info lsass
```

.lwsm get-log <service> [<facility>]

Lists the logging state given the service and optionally the facility.

Example:

```
/opt/pbis/bin/lwsm get-log gpagent
```

Example output:

```
<default>: syslog LOG_DAEMON at DEBUG
```

./lwsm set-log-target [-p, -- persist] <service> <facility> <type> [<target> | <syslog facility>]

Set the log target for a given service and facility. Optionally include **-p, --persist** to save the log type and target so they will be used when the service starts. The facility is the tag of an AD Bridge facility or a dash (-). Supported types are **none**, **syslog**, and **file**.

Example:

```
/opt/pbis/bin/lwsm set-log-target lsass daemon syslog
```

A type of **syslog** uses the **LOG_DAEMON** facility by default. You can override this by setting a syslog facility name.

Example:

```
/opt/pbis/bin/lwsm set-log-target lsass - syslog local3
```

A type of **file** directs logging to a file and requires you to set a target, such as file name.

Example:

```
/opt/pbis/bin/lwsm set-log-target lsass - file log.txt
```

./lwsm set-log-level [-p, --persist] <service> <facility> <level>

Set the log level for a given service and facility. Supported levels are **error**, **warning**, **info**, **verbose**, **debug**, and **trace**. The default setting is **error**.

Optionally include **-p, --persist** to save the log level so it will be used when the service starts. The log level is changed only until the authentication service (lsass) or the computer restarts. Syslog messages are logged through the daemon facility.

Example:

```
/opt/pbis/bin/lwsm set-log-level lsass - DEBUG
```

Note: *You cannot use* ***--persist*** *with the service* ***lwsmd****, that is, where the* ***<service>*** *value is a dash (****-****). To persist the settings for* ***lwsmd****, you must change the settings in the startup script. The script name and location depends on the platform and is either* ***/etc/init.d/lwsmd*** *or* ***/etc/systemd/system/lwsmd.service****.*

./lwsm reset-log-defaults <service>

Clear any saved log level, type, and target default values. This does not affect the service's current log settings. You must restart the service to get the new default values.

Example:

```
 /opt/pbis/bin/lwsm reset-log-default lsass
```

Note: *You cannot use* ***reset-log-defaults*** *with the service* ***lwsmd****, that is, where the* ***<service>*** *value is a dash (****-****).*

./lwsm tap-log <service> <facility> <level>

Temporarily redirect logging for the given service and facility to **stdout** with the given log level.

Example:

```
/opt/pbis/bin/lwsm tap-log lsass - WARNING
```

./lwsm gdb <service>

Attach **gdb** to the specified running service.

Example:

```
/opt/pbis/bin/lwsm gdb lsass
```

AD Bridge Command Line Reference

This chapter provides an overview of the commands in AD Bridge. Most of the commands are intended to be run as root.

> *For information about troubleshooting the Group Policy commands for AD Bridge Enterprise, please see the Group Policy Reference Guide at www.beyondtrust.com/docs/ad-bridge/how-to/group-policy.*

Change the Hostname in the Local Provider (set-machine-name)

After you change the hostname of a computer, you must also change the name in the AD Bridge local provider database so that the local AD Bridge accounts use the correct prefix.

```
./lsa set-machine-name <hostname>
```

> ***Note:*** *Run the command as root.*

List the Status of Authentication Providers (get-status)

AD Bridge includes two authentication providers: a local provider and an Active Directory provider. If the AD provider is offline, you cannot log on with your AD credentials. You can check the status of the authentication providers.

```
/opt/pbis/bin/get-status
```

Healthy result output:

```
LSA Server Status:
Agent version: 5.4.0
        Uptime:   22 days 21 hours 16 minutes 29 seconds
[Authentication provider: lsa-local-provider]
        Status:   Online
        Mode:     Local system
[Authentication provider: lsa-activedirectory-provider]
        Status:   Online
        Mode:     Un-provisioned
        Domain:   example.com
        Forest:   example.com
        Site:     Default-First-Site-Name
```

Unhealthy result output:

An unhealthy result will not include the AD authentication provider or will indicate that it is offline. If the AD authentication provider is not listed in the results, restart the authentication service.

If the result looks like the line below, check the status of the AD Bridge Enterprise services to make sure they are running.

Failed to query status from LSA service. The LSASS server is not responding.

To check the status of the services. Run the following command as root:

```
/opt/pbis/bin/lwsm list
```

List the Domain (ad-get-machine)

Retrieve the Active Directory domain to which the computer is connected.

```
./lsa ad-get-machine account
```

List Domain Controllers (get-dc-list)

List the domain controllers for a target domain. You can delimit the list in several ways, including by site.

```
./get-dc-list
```

Example:

```
[root@rhel5d bin]# ./get-dc-list example.com
Got 1 DCs:
===========
DC 1: Name = 'steveh-dc.example.com', Address = '192.168.100.132'
```

List Domain Controller Information (get-dc-name)

Display the name of the current domain controller for the domain you specify. This command can help you select a domain controller.

```
./get-dc-list
```

Example: *To select a domain controller, run the following command as root until the domain controller you want is displayed.*

```
/opt/pbis/bin/get-dc-name <domain-name> --force
```

List Domain Controller Time (get-dc-time)

Displays the time of the current domain controller for the domain that you specify. This command can help you determine whether there is a Kerberos time-skew error between a client and a domain controller.

```
./get-dc-time
```

Example:

```
[root@rhel5d bin]# ./get-dc-time
example.com DC TIME: 2009-09-08 14:54:18 PDT
```

List Computer Account Information (lsa ad-get-machine)

Print out the computer account name, computer account password, SID, and other information by running the following command as root.

```
./lsa ad-get-machine account <domain-name>
```

Example:

```
/opt/pbis/bin/lsa ad-get-machine account example.com
```

Dynamically Update DNS (update-dns)

Registers an IP address for the computer in DNS. The command is useful when you want to register A and PTR records for your computer and the DHCP server is not registering them.

```
./update-dns
```

Example: *Register an IP address:*

```
/opt/pbis/bin/update-dns --ipaddress 192.168.100.4 --fqdn corp.example.com
```

If your system has multiple NICs and you are trying to register all their IP addresses in DNS, run the command once with multiple instances of the option:

```
/opt/pbis/bin/update-dns --fqdn corp.example.com --ipaddress 192.168.100.4 --ipaddress
192.168.100.7 --ipaddress 192.168.100.9
```

To troubleshoot, add the option with the parameter:

```
/opt/pbis/bin/update-dns --loglevel debug --fqdn corp.example.com --ipaddress
192.168.100.4 --ipaddress 192.168.100.7
```

Note: --fqdn *is the fully qualified domain name for the client computer.*

Manage the AD Cache (ad-cache)

This command manages the AD Bridge cache for Active Directory users and groups on Linux and Unix computers. You can use the command to clear the cache. The command's arguments can delete from the cache a user, a group, or all users and groups.

```
./ad-cache
```

***Example:** Delete all the users and groups from the cache.*

```
/opt/pbis/bin/ad-cache --delete-all
```

To reclaim disk space from SQLite after you clear the cache when you are using the non-default SQLite caching option, execute the following command as root, replacing with your fully qualified domain name:

```
/opt/pbis/bin/sqlite3 /var/lib/pbis/db/lsass-adcache.filedb.fqdn vacuum
```

You can also use the command to enumerate users in the cache, which may be helpful in troubleshooting.

```
[root@rhel5d bin]# ./ad-cache --enum-users
TotalNumUsersFound:      0
[root@rhel5d bin]# ssh example.com\\hab@localhost
Password:
Last login: Tue Aug 11 15:30:05 2009 from rhel5d.example.com
[EXAMPLE\hab@rhel5d ~]$ exit
logout
Connection to localhost closed.
[root@rhel5d bin]# ./ad-cache --enum-users
User info (Level-0):
====================
Name:      EXAMPLE\hab
Uid:       593495196
Gid:       593494529
Gecos:     <null>
Shell:     /bin/bash
Home dir: /home/EXAMPLE/hab
TotalNumUsersFound:      1
[root@rhel5d bin]#
```

Copy Files Across Disparate Operating Systems (lwio-copy)

This command lets you copy files across computers running different operating systems. For example, you can copy files from a Linux computer to a Windows computer.

```
lwio-copy
```

There are two prerequisites to use lwio-copy:

- The **lwio** service must be running.
- The **rdr** driver must be available as specified by the registry. By default, the **rdr** driver is available at **/opt/pbis/lib/lwio-driver/rdr.so**.

Domain Join Tool Commands for AD Bridge

The command-line utility **domainjoin-cli** gives you tools to add or remove accounts from a domain. The utility prompts for the domain, username, and organizational unit parameters. A history of entries is saved between join and leave prompts. To access the utility, run the following command:

```
/opt/pbis/bin/domainjoin-cli
```

Options

--help

Displays the command-line options and commands.

Example:

```
domainjoin-cli --help
```

--help-internal

Displays a list of the internal debugging and configuration commands.

Example:

```
domainjoin-cli --help-internal
```

--logfile {.| path}

Generates a log file or prints the log to the console.

Example:

```
domainjoin-cli --logfile /var/log/domainjoin.log join example.com Administrator
```

```
domainjoin-cli --logfile . join example.com Administrator
```

--loglevel {error|warning|info|verbose}

Adjusts the logging details generated during a domain join.

Example:

```
domainjoin-cli --loglevel error
```

Join Commands

query

Displays the hostname, current domain, and distinguished name, which includes the organizational unit to which the computer belongs. If the computer is not joined to a domain, it displays only the hostname.

Example:

```
domainjoin-cli query
```

setname computerName

Renames the computer and modifies the **/etc/hosts** file with the name that you enter. **computerName** is a required field. If not provided, you are prompted for it.

Example:

```
domainjoin-cli setname RHEL44ID
```

join [--ou organizationalUnit] domainName userName

Joins the computer to the domain. If not provided, you are prompted to enter the domain, username, and password.

You can use the **--ou** option to join the computer to a specific organizational unit in the domain by setting the path to the OU. The path to the OU is top down and separated by a slash (/). To be prompted for an organizational unit you must pass in **--ou**. When you use this option, you must use an account that is a member in the Domain Administrators security group.

Example:

```
domainjoin-cli join --ou Eng/Dev example.com Administrator
```

join --notimesync

Joins the computer to the domain without synchronizing the computer's time with the domain controller's.

When you use this option, the **sync-system-time** value for lsass is set to **no**.

Example:

```
domainjoin-cli join --notimesync example.com Administrator
```

join --trustEnumerationWaitSeconds 60

The length of time lsass waits for trust enumeration to finish during startup. The range is 1 - 1000 seconds.

Example:

```
domainjoin-cli join --trustEnumerationWaitSeconds 300
```

Leave Commands

leave [userName]

Removes the computer from the Active Directory domain. If the username is provided, the computer account is disabled in Active Directory. If not provided, you are prompted to enter a username and password.

Example:

```
domainjoin-cli leave
```

```
domainjoin-cli leave smithy@example.com
```

leave [--enable <module> | --disable <module>]

Enables or disables the module when you run the leave command.

Example:

```
domainjoin-cli leave --enable pam
```

leave [--keepLicense]

Retains the license information after the computer leaves the domain. The license key is released by default when you run the leave command.

Example:

```
domainjoin-cli leave --keepLicense
```

leave [--deleteAccount <user name> [<password>]]

Deletes the computer account after the computer leaves the domain.

Example:

```
domainjoin-cli leave --deleteAccount Administrator AdminPassword
```

leave [--advanced] --preview [username] [password]

Displays information on the configuration.

Example:

```
domainjoin-cli leave --advanced --preview Administrator AdminPassword
```

leave --details <module>

Displays the configuration information for the module.

Example:

```
domainjoin-cli leave --details pam
```

Join Mode Commands

--assumeDefaultCell { auto | no | force }

In **Assume Default Cell** mode, information is not read from the cells, but from the user objects and group objects directly. This supports joining to a domain which does not have any named or default cells.

If set to **auto**, enable this mode when no cells are found.

If set to **force**, enable this mode even if named or default cells exist. When this mode is enabled, **get-status** reports the AD authentication provider mode as **Default Cell (Assumed)**.

The default setting is **no**.

***Note:** This mode is intended for Proof of Concept (PoC) and small environments. It does not require cells or schema changes. User and group information is read directly from the domain controllers in the forest; no Global Catalog searches are used. Features that rely on items stored in the cell (for example, custom NIS maps) are not supported in this mode.*

Example:

```
domainjoin-cli join --assumeDefaultCell auto
```

--unprovisioned { auto | no | force }

When set, the AD provider computes the user and group IDs from their security identifier. It uses local settings for the Unix shell and home directory, ignoring the values set in AD.

If set to **auto**, enable this mode when no cells are found.

If set to **force**, enable this mode even if named or default cells exist. The default setting is **no**.

Example:

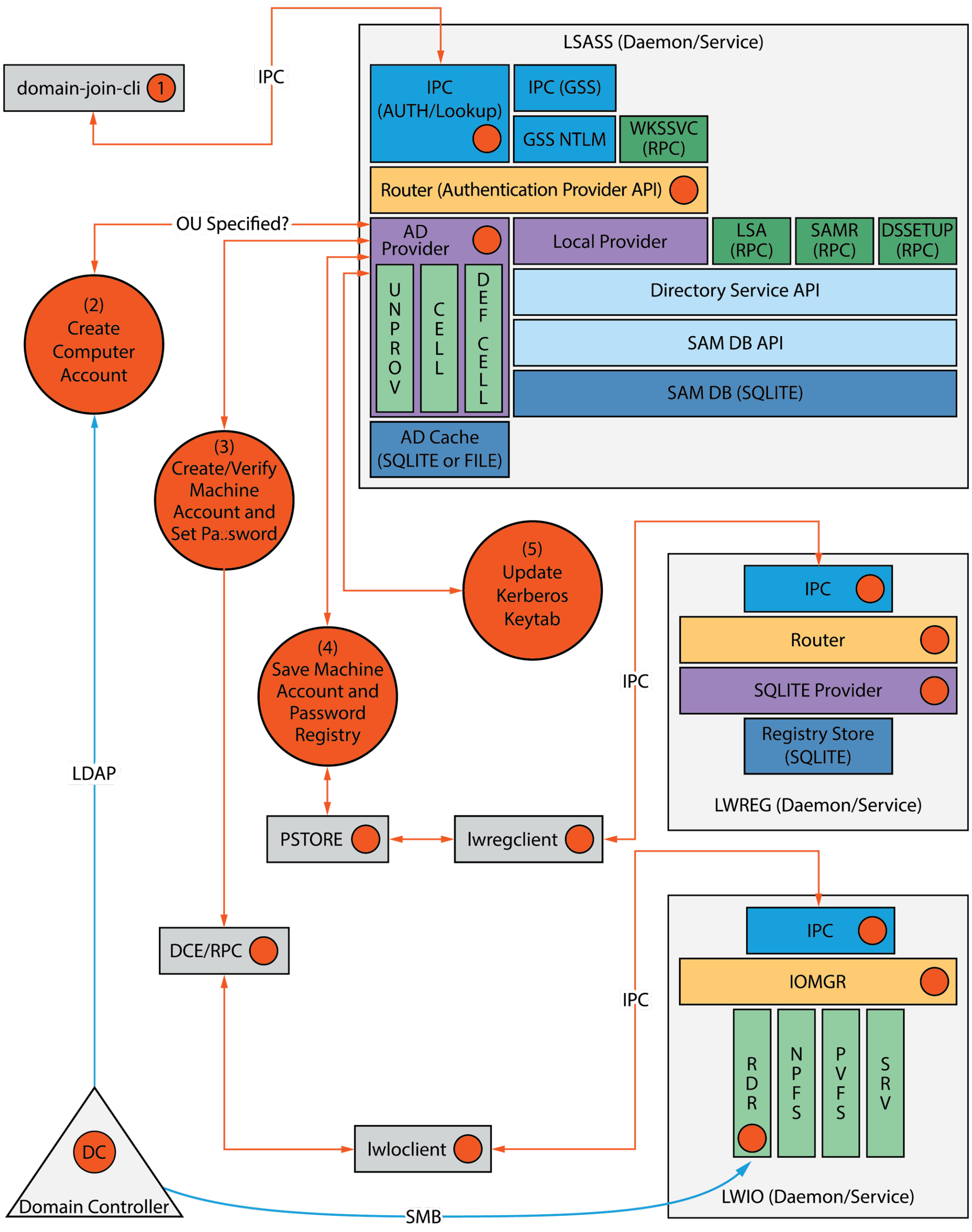
```
domainjoin-cli join --unprovisioned force
```

Domain Join Advanced Commands for AD Bridge

The advanced commands in this section can be used to troubleshoot issues when configuring a Linux or Unix computer.

Review the *Domain Join Dataflow* diagram to see how systems interact when you join a domain.

Domain Join Component Interaction



Preview the Stages of the Domain Join for Your Computer

Preview the stages of the domain join for a computer, including the domain stage, DNS name stage, and configuration stage that are run after you start the process.

Example:

```
[root@rhel4d bin]# domainjoin-cli join --preview example.com
Joining to AD Domain:   example.com
With Computer DNS Name: rhel4d.example.com

The following stages are currently configured to be run during the domain join:
join     - join computer to AD
krb5     - configure krb5.conf
nsswitch - enable/disable AD Bridge nsswitch module
start    - start daemons
pam      - configure pam.d/pam.conf
ssh      - configure ssh and sshd
```

Check Required Configurations with Join Command

List the modules that apply to your operating system, when joining a domain, including those modules that will not be run.

Example:

```
domainjoin-cli join --advanced --preview example.com
```

Check Required Configurations with Leave Command

List the modules that apply to your operating system when leaving a domain, including those modules that will not be run.

Example:

```
domainjoin-cli leave --advanced --preview example.com
```

Example:

```
[root@rhel4d bin]# domainjoin-cli join --advanced --preview example.com
Joining to AD Domain:   example.com
With Computer DNS Name: rhel4d.example.com
[X] [F] stop              - stop daemons
    [F] hostname          - set computer hostname
    [F] keytab            - initialize kerberos keytab
[X] [N] join              - join computer to AD
[X] [N] nsswitch          - enable/disable AD Bridge nsswitch module
[X] [N] cache             - manage caches for this host
[X] [N] start             - start daemons
[X] [N] krb5              - configure krb5.conf
```

```
[X] [N] pam               - configure pam.d/pam.conf
[X] [S] ssh               - configure ssh and sshd
    [F] DDNS              - Configure Dynamic DNS Entry for this host

Key to flags
[F]ully configured        - the system is already configured for this step
[S]ufficiently configured - the system meets the minimum configuration requirements for
this step
[N]ecessary               - this step must be run or manually performed.
[X]                       - this step is enabled and will make changes
[ ]                       - this step is disabled and will not make changes
```

Modules

The AD Bridge Enterprise domain join tool includes the following modules, which are the components and services that the tool must configure before it can join a computer to a domain:

Module	Description
join	Joins the computer to Active Directory
leave	Deletes the machine account in Active Directory
stop	Stops services so that the system can be configured
start	Starts services after configuration
firewall	Opens ports to the domain controller
hostname	Sets the computer hostname
krb5	Configures **krb5.conf**
pam-mode	Switches authentication from LAM to PAM
nsswitch	Enables or disables the AD Bridge Enterprise **nsswitch** module
pam	Configures **pam.d** and **pam.conf**
lam-auth	Configures LAM for Active Directory authentication
ssh	Configures **ssh** and **sshd**
bash	Fixes the bash prompt for backslashes in usernames
gdm	Fixes the **gdm** pre-session script for spaces in usernames

Join and Leave Commands for the Modules

domainjoin-cli join --advanced --preview domainName

View the modules that must be configured on your computer.

Example:

```
domainjoin-cli join --advanced --preview example.com
```

domainjoin-cli join --details module domainName joinAccount

View more information about a module, including the modules that are configured.

Example:

```
domainjoin-cli join --details nsswitch example.com Administrator
```

domainjoin-cli join --disable module domainName accountName

Turn off a module when you join a domain. Disabling a module can be useful in cases where a module has been manually configured or in cases where you must ensure that certain system files will not be modified.

Note: *If you disable a necessary module and you have not manually configured it, the domain join utility will not join your computer to the domain.*

Example:

```
domainjoin-cli join --disable nsswitch example.com Administrator
```

domainjoin-cli join --enable module domainName accountName

Turn on a module when you join a domain.

Example:

```
domainjoin-cli join --enable nsswitch example.com Administrator
```

domainjoin-cli leave --advanced --preview domainName

List the modules that apply to your operating system when leaving a domain, including those modules that will not be run.

Example:

```
domainjoin-cli leave --advanced --preview example.com
```

domainjoin-cli leave --details module domainName joinAccount

View more information about a module, including the modules that are configured.

Example:

```
domainjoin-cli leave --details pam example.com Administrator
```

domainjoin-cli leave --disable module domainName accountName

Turn off a module when you leave a domain.

Example:

```
/opt/pbis/bin/domainjoin-cli leave --advanced --preview --disable nsswitch example.com
[X] [N] nsswitch          - enable/disable  nsswitch module

[F] DDNS - Configure Dynamic DNS Entry for this host
[X] [S] ssh - configure ssh and sshd
[F] pam - configure pam.d/pam.conf
[F] nsswitch - enable/disable nsswitch module
[F] krb5 - configure krb5.conf
[F] stop - stop daemons
[F] leave - leave the domain and release the license
[F] keytab - initialize kerberos keytab

Key to flags
[F]ully configured - the system is already configured for this step
[S]ufficiently configured - the system meets the minimum configuration
requirements for this step
[N]ecessary - this step must be run or manually performed.

[X] - this step is enabled and will make changes
[ ] - this step is disabled and will not make changes
```

Configuration and Debugging Commands

The **domainjoin-cli** tool includes commands for debugging the **domain-join** process and for configuring or preconfiguring a module.

For example, run the **configure** command to preconfigure a system before you join a domain, a useful strategy when you are deploying AD Bridge Enterprise in a virtual environment and you need to preconfigure the **nsswitch**, **ssh**, or **pam** module of the target computers to avoid restarting them after they are added to the domain.

Note: *The **--testprefix** option supports testing system configuration file changes. If supplied, the **--testprefix** directory is prepended to the path of the configuration file target.*

*For example, the following command changes the **/testconfig/etc/nsswitch.conf** file instead of **/etc/nsswitch.conf**:*

```
configure --enable --testprefix testconfig nsswitch
```

Example: *Example with nsswitch*

```
domainjoin-cli configure --enable nsswitch
```

Example: *Example with fixfqdn*

```
domainjoin-cli fixfqdn
```

Help Syntax:

```
domainjoin-cli --help-internal
```

```
fixfqdn
configure { --enable | --disable } [--testprefix <dir>] pam
configure { --enable | --disable } [--testprefix <dir>] nsswitch
configure { --enable | --disable } [--testprefix <dir>] ssh
configure { --enable | --disable } [--testprefix <dir>] [--long <longdomain>] [--short
<shortdomain>] krb5
configure { --enable | --disable } eventfwdd
configure { --enable | --disable } reapsysld
get_os_type
get_arch
get_distro
get_distro_version
```

User and Group Commands in AD Bridge

Find a User or a Group

You can check a domain user's or group's information by either name or ID. These commands can verify that the client can locate the user or group in Active Directory.

Find a User by Name

```
find-user-by-name domain\\username
```

Search for a user by name.

***Note:** Replace **domain\\username** with the full domain user name or the single domain user name of the user.*

Example:

```
/opt/pbis/bin/find-user-by-name mydomain\\trejo
```

Optionally set the level of detail of information that is returned.

Example:

```
/opt/pbis/bin/find-user-by-name --level 2 mydomain\\trejo
User info (Level-2):
====================
Name:                          trejo
SID:                           S-1-5-21-3447809367-3151979076-456401374-1135
UPN:                           trejo@MYDOMAIN.EXAMPLE.COM
Generated UPN:                 NO
DN:                            CN=trejo,CN=Users,DC=MYDOMAIN,DC=EXAMPLE,DC=COM
Uid:                           239600751
Gid:                           239600770
Gecos:                         Markus Trejo
Shell:                         /bin/sh
Home dir:                      /home/MYDOMAIN/trejo-macbook/trejo-bvt
LMHash length:                 0
NTHash length:                 0
Local User:                    NO
Account disabled (or locked): FALSE
Account expired:               FALSE
Password never expires:        TRUE
Password Expired:              FALSE
Prompt for password change:    YES
```

```
User can change password:     YES
Days till password expires:   0
Logon restriction:            NO
trejo-macbook:~ root#
```

Find a User by User ID

```
find-user-by-id UID
```

Search for a user by UID.

Example:

```
/opt/pbis/bin/find-user-by-id 593495196
```

Find a User in Active Directory by Security Identifier

```
find-by-sid SID
```

Find a user in Active Directory by security identifier (SID).

Note: *Run the command as root.*

Example:

```
/opt/pbis/bin/find-user-by-id 593495196
```

```
[root@rhel4d bin]# /opt/pbis/bin/find-by-sid S-1-5-21-382349973-3885793314-468868962-1180
User info (Level-0):
====================
Name:     EXAMPLE\hab
SID:      S-1-5-21-382349973-3885793314-468868962-1180
Uid:      593495196
Gid:      593494529
Gecos:    Jurgen Habermas
Shell:    /bin/ sh
Home dir: /home/ EXAMPLE/ hab
```

Find a Group by Name

```
find-group-by-name domain\\groupname
```

Finds a group.

Example:

```
/opt/pbis/bin/find-group-by-name example.com\\dnsadmins
```

Find a Group by ID

```
find-group-by-id GID
```

Finds a group using the group ID.

Example:

```
/opt/pbis/bin/find-group-by-id 593494534
```

```
[root@rhel4d bin]# /opt/pbis/bin/find-group-by-id 593494534
Group info (Level-0):
====================
Name:     EXAMPLE\schema^admins
Gid:      593494534
SID:      S-1-5-21-382349973-3885793314-468868962-518
```

List Users or Groups

List Users

```
enum-users
```

Enumerate the users in Active Directory and view their members, group IDs, and security IDs. The AD Bridge agent enumerates users in the primary domain. Users in trusted domains and linked cells are not enumerated. NSS membership settings in the registry do not affect the result of the command.

Note: *To view full information about the users, include the level option when you execute the command:* ***/opt/pbis/bin/enum-users --level 2****.*

Example:

```
/opt/pbis/bin/enum-users
```

```
User info (Level-2):
====================
Name:                       EXAMPLE\sduval
UPN:                        SDUVAL@EXAMPLE.COM
Generated UPN:              NO
Uid:                        593495151
Gid:                        593494529
Gecos:                      Shelley Duval
Shell:                      /bin/sh
Home dir:                   /home/EXAMPLE/sduval
LMHash length:              0
NTHash length:              0
Local User:                 NO
Account disabled:           FALSE
Account Expired:            FALSE
Account Locked:             FALSE
Password never expires:     FALSE
Password Expired:           FALSE
Prompt for password change: NO
```

List Members

```
enum-members
```

Enumerate the members of a group. This command can return user or group information if they are part of the group specified.

If there are nested groups and the user runs the command **/opt/pbis/bin/enum-members --group --by-name <domain name>\\<group name>**, it will return the nested groups. If the user runs the command **/opt/pbis/bin/enum-members --user --by-name <domain name>\\<group name>**, it will return the users in that group.

Example:

```
/opt/pbis/bin/enum-members
```

Example output for users returned in a group:

```
User object (1] (5-1-5-21-3705731645-4233351989-3429207207-1127)
Enabled: yes
Distinguished name: CN=user,OU=thirdfloor,DC=mydomain,DC=com SAM account name: User
NetBIOS domain name: mydomain UPN: user@mydomain.com Display Name: User
Alias: <null>
UNIX name: mydomain\User GECOS: User
Shell: /bin/sh
```

```
Home directory: /home/local/mydomain/User
Windows home directory: <null> Local windows home directory: UID: 822608999
Primary group SID: S-1-5-21-3705731645-4233351989-3429207207-513 Primary GID: 822608385
Password expired: no
Password never expires: no
Change password on next logon: no User can change password: yes Account disabled: no
Account expired: no
Account locked: no

User object (2] (5-1-5-21-3705731645-4233351989-3429207207-1126)
Enabled: yes
Distinguished name: CN= user,0U= thirdfloor,DC=mydomain,DC=com SAM account name: User
NetBIOS domain name: mydomain UPN: mydomain.com Display Name: User
Alias: <null>
UNIX name: mydomain\User GECOS: User
Shell: /bin/sh
Home directory: /home/local/mydomain/User
Windows home directory: <null> Local windows home directory: UID: 822608998
Primary group SID: S-1-5-21-3705731645-4233351989-3429207207-513 Primary GID: 822608385
Password expired: no
Password never expires: no
Change password on next logon: no User can change password: yes Account disabled: no
Account expired: no
Account locked: no

User object (3) (5-1-5-21-3705731645-4233351989-3429207207-1125)
Enabled: yes
Distinguished name: CN= user,0U=thirdfloor,DC=mydomain,DC=com SAM account name: User
NetBIOS domain name: mydomain UPN: user@mydomain.com Display Name: User
Alias: <null>
UNIX name: mydomain\user GECOS: User
Shell: /bin/sh
Home directory: /home/local/mydomain/user
```

List Groups

```
enum-groups
```

Enumerate the groups in Active Directory and view the group IDs and security IDs of members. The AD Bridge agent enumerates groups in the primary domain. Groups in trusted domains and linked cells are not enumerated. NSS membership settings in the registry do not affect the result of the command.

> ***Note:*** *To view full information about the groups, include the level option when you execute the command:* ***/opt/pbis/bin/enum-users --level 2****.*

Example:

```
/opt/pbis/bin/enum-groups
```

List Groups for a User

You can list the groups where a particular user is a member.

```
list-groups-for-user
```

List the groups where a particular user is a member. You can search either by user name or user ID.

Example:

```
/opt/pbis/bin/list-groups-for-user  --uid 593495196
```

```
[root@rhel5d bin]# ./list-groups-for-user example\\hab
Number of groups found for user 'example\hab' : 2
Group[1 of 2] name = EXAMPLE\enterprise^admins (gid = 593494535)
Group[2 of 2] name = EXAMPLE\domain^users (gid = 593494529)
```

Local Accounts Commands in AD Bridge

The AD Bridge Enterprise local authentication provider for local users and groups includes a full local authentication database. With functionality similar to the local SAM authentication database on every Windows computer, the local authentication provider lets you create, modify, and delete local users and groups on Linux and Unix computers by using the following commands.

To execute the commands that modify local accounts, you must use either the root account or an account that has membership in the local administrators group. The account can be an Active Directory account if you manually add it to the local administrators group. For example, you can add the Domain Administrators security group from Active Directory to the local administrators group, and then use an account with membership in the Domain Administrators security group to execute the commands.

***Note:** To authenticate a local provider user before the machine is joined to a domain, you must run the following commands to enable pam and nsswitch:*

```
/opt/pbis/bin/domainjoin-cli configure --enable nsswitch
/opt/pbis/bin/domainjoin-cli configure --enable pam
/opt/pbis/bin/config Providers "ActiveDirectory" "Local"
```

Command	Description
add-user	Adds a user to the local authentication database.
add-group	Adds a group member to the local authentication database.
del-user	Deletes a user from the local authentication database.
del-group	Deletes a group from the local authentication database.
mod-user	Modifies a user's account settings in the local authentication database, including an account's expiration date and password. You can also enable a user, disable a user, unlock an account, or remove a user from a group.
mod-group	Adds members to or removes members from a group in the local authentication database. `/opt/pbis/bin/mod-group --add- members DOMAIN\\Administrator BUILTIN\\Administrators`

Add Domain Accounts to Local Groups

You can add domain users to your local groups on a Linux or Unix computer by placing an entry for the user or group in the **/etc/group** file. Adding an entry for an Active Directory user to your local groups can give the user local administrative rights. The entries must adhere to the following rules:

- Use the correct case; entries are case sensitive.
- Use a user or group's alias if the user or group has one in Active Directory.
- If the user or group does not have an alias, set the user or group in the AD Bridge Enterprise canonical name format of **NetBIOSdomainName\sAMAccountName**.

***Note:** For users or groups with an alias, the AD Bridge Enterprise canonical name format is the alias, which you must use; you cannot use the format of **NetBIOS domain name\SAM account name**.*

For users and groups without an alias, the form of an entry is as follows:

```
root:x:0:EXAMPLE\kristeva
```

For users and groups with an alias, the form of an entry is as follows:

```
root:x:0:kris
```

In **/etc/group**, the slash character separating the domain name from the account name does not typically need to be escaped.

***Tip:** On Ubuntu, you can give a domain user administrative privileges by adding the user to the admin group as follows:*

```
admin:x:119:EXAMPLE\bakhtin
```

Check a User's Canonical Name on Linux

To determine the canonical name of an AD Bridge user on Linux, execute the following command, replacing the domain and user in the example with your domain and user:

```
getent passwd example.com\\hab
EXAMPLE\hab:x:593495196:593494529: Jurgen Habermas:/home/local/ EXAMPLE/ hab:/bin/ sh
```

In the results, the user's AD Bridge Enterprise canonical name is the first field.

Extend File Mode Permissions with POSIX ACLs

When you have to grant multiple users or groups access to a file, directory, or Samba share on a Linux server, you can use POSIX access control lists to extend the standard file mode permissions.

Linux and Unix file mode permissions control access only for a single user, a single group, and then everyone else. Thus, the only means of granting access to more than one group with the standard file modes is either to nest the groups together or to give everyone access, approaches that are often unacceptable. Nested groups can be a maintenance burden, and granting access to everyone can undermine security. As for Samba shares, it is insufficient to add multiple users and groups to the **valid users** parameter in **smb.conf** if the underlying file system does not allow them access.

Prerequisites:

You must have the **acl** package installed. You can determine this as follows:

```
# rpm – qa | grep acl
libacl-2.2.23-5
acl-2.2.23-5
```

The file system must be mounted with **acl** in the option list. You can determine this using the **mount** command:

```
# mount
/dev/sda1 on / type ext3 (rw,acl)
```

As shown above, the root file system has been mounted with read-write (**rw**) and **acl** options. If you do not see **acl** in the options for the file system you are working with, modify **/etc/fstab** to include this option, and then remount the file system. In the case of the root file system, you may need to restart the system.

All users and groups must be created before adding them to the ACL. In the case of Active Directory users, they must be preceded by the domain unless user aliases have to be configured (for example, DOMAIN\username).

Example: *This example uses a directory called **testdir**. The process is the same for files.*

Here are the standard file mode permissions of the testdir directory.

```
[aciarochi@rhel4-devel tmp]$ ls -ld testdir
drwxrwx---  2 root root 4096 Dec 14 13:28 testdir
```

*You can view the extended ACL using the **getfacl** utility. In this case, it shows the same information, in a different format:*

```
[aciarochi@rhel4-devel tmp]$ getfacl testdir
# file: testdir
# owner: root
# group: root
user::rwx
group::rwx
other::---
```

*With these permissions, only the root user and members of the root group are allowed to open the directory. Since the **aciarochi** user is not in the root group, they are denied access:*

```
[aciarochi@rhel4-devel tmp]$ cd testdir
-bash: cd: testdir: Permission denied
```

*However, we can grant access to **aciarochi** by using the **setfacl** utility to add them to the ACL. We must switch to the root user, since that is the directory owner. Once the ACL is set, **aciarochi** can open the directory:*

```
[root@rhel4-devel ~]# setfacl -m u:aciarochi:rwx /tmp/testdir/
[root@rhel4-devel ~]# exit
logout
[aciarochi@rhel4-devel tmp]$ cd testdir
[aciarochi@rhel4-devel testdir]$ pwd
/tmp/testdir
```

*Notice that the standard file mode permissions have not changed, except for the addition of a plus (**+**) at the end, indicating that extended file permissions are in effect:*

```
[aciarochi@rhel4-devel tmp]$ ls -ld /tmp/testdir/
drwxrwx---+ 2 root root 4096 Dec 14 13:28 /tmp/testdir/
```

*Additional groups can be added in the same manner, using a **g:** instead of a **u:** to indicate a group. In the following example, we grant read and execute (open) access to the **ftp** group:*

```
[root@rhel4-devel ~]# setfacl -m g:ftp:r-x /tmp/testdir
[root@rhel4-devel ~]# getfacl testdir
# file: testdir
# owner: root
# group: root
user::rwx
user:aciarochi:rwx
group::rwx
group:ftp:r-x
mask::rwx
other::---
```

Using POSIX ACLs to Grant AD Accounts Access to Subversion

With AD Bridge, you can use AD accounts with Subversion. Use POSIX ACLs to give a domain group write access to the SVN repository.

***Note:** Use only one forward slash (/) in **/etc/group**. The entry is case-sensitive. The domain name must be uppercase and the username lowercase.*

Example:

```
$ svnadmin create /data/foo

## Add domain admins to the default directory ace
$ find /data/foo -type d | xargs setfacl -d -m “g:AD\domain^admins:rwx”

## Add domain admins to the directory ace
$ find /data/foo -type d | xargs setfacl -m “g:AD\domain^admins:rwx”

## Add domain admins to the ace for files
$ find /data/foo -type f | xargs setfacl -m “g:AD\domain^admins:rw”

$ getfacl /data/foo
# file: foo
# owner: AD\134gjones
# group: AD\134unixusers
user::rwx
group::r-x
group:AD\134domain^admins:rwx
mask::rwx
other::r-x
default:user::rwx
default:group::r-x
default:group:AD\134domain^admins:rwx
```

```
default:mask::rwx
default:other::r-x
```

Active Directory Tool Commands in AD Bridge

AD Bridge Enterprise includes a tool to modify objects in Active Directory. Using the tool, you can:

- Query and modify objects in Active Directory.
- Find and manage objects in AD Bridge Cells.

Command:

```
/opt/pbis/bin/adtool
```

Help Syntax:

```
/opt/pbis/bin/adtool --help -a
```

Active Directory Commands

add-to-group

Add a domain user, group, or computer to a security group.

Add TestUser to TestGroup:

```
adtool -a add-to-group --user TestUser --to-group=TestGroup
```

Add TestGroup2 to TestGroup:

```
adtool -a add-to-group --group TestGroup2 --to-group=TestGroup
```

Add TestComputer3 to TestGroup:

```
adtool -a add-to-group --computer TestComputer3 --to-group=TestGroup
```

delete-object

Delete an object.

Delete a cell object and all its children if any (--force):

```
adtool -a delete-object --dn OU=TestOU --force
```

disable-user

Disable a user account in Active Directory.

```
/opt/pbis/bin/adtool -a disable-user --name=user6
```

enable-user

Enable a user account in Active Directory.

```
adtool -a enable-user --name=TestUser
```

unlock-account

Unlock a user or computer account.

```
adtool -a unlock-account --user=aduser
```

lookup-object

Retrieve object attributes.

```
adtool -a lookup-object --dn=CN=RHEL7,CN=Computers,DC=company,DC=com
```

move-object

Move and rename an object.

Rename AD object OU=OldName and move it to a new location:

```
adtool -a move-object --from OU=OldName,DC=department,DC=company,DC=com --to
OU=NewName,OU=TestOU,DC=department,DC=company,DC=com
```

new-computer

Create a computer object.

```
/opt/pbis/bin/adtool -d domain.com -n pbisadmin -x Password1 -a new-computer --dn
OU=Test,DC=domain,DC=com --name=tst-QAmachine
```

new-computer --keytab

Create a computer object with a keytab file. An additional option **--spn** can be used to set the Service Principal Name on the computer object.

```
adtool -a new-computer --dn "CN=<computers>,DC=<domain>,DC=<NET>" --name <ACCOUNT NAME> --
password <PASSWD> --keytab-file <file.keytab> --spn="HOST, NFS"
```

new-group

Create a global security group.

Create a new group in OU=Groups,OU=TestOu:

```
adtool -a new-group --dn OU=Groups,OU=TestOu --name TestGroup
```

new-ou

Create an organizational unit.

Create an OU in a root naming context:

```
adtool -a new-ou --dn OU=TestOu
```

Create an OU in DC=department,DC=company,DC=com:

```
adtool -a new-ou --dn OU=TestOu,DC=department,DC=company,DC=com
```

Create an AD Bridge Cell in the OU TestOU setting the default login shell property to /bin/ksh:

```
adtool -a new-ou --dn OU=TestOu --default-login-shell=/bin/ksh
```

new-user

Create a user account.

Create an account for TestUser in OU=Users,OU=TestOu:

```
adtool -a new-user --dn OU=Users,OU=TestOu --cn=TestUserCN --logon-name=TestUser --
password=$PASSWD
```

new-user --keytab

Create a user account with a keytab file. An additional option **--spn** can be used to set the Service Principal Name on the user object.

If there is no keytab with an existing account, then password changes that include a keytab location will create a keytab file for that account:

```
adtool -a reset-user-password --name <USERNAME> --password <PASSWD> --keytab-file
/tmp/file.keytab --spn="NFS" --no-must-change-password
```

If a keytab exists for an account, then password changes are added to the current keytab.

Note:** To create a keytab file, you must use: **--no-must-change-password --account-enabled

```
adtool  -a new-user --dn "OU=<OU>,DC=<DOMAIN>,DC=<NET>" --logon-name <USER NAME>  --first-name
<FIRSTNAME> --last-name <LAST NAME> --password <PASSWD> --keytab-file /tmp/file.keytab --
spn="NFS" --no-must-change-password --account-enabled
```

remove-from-group

Remove a user, group, or computer from a security group.

Remove TestUser from TestGroup:

```
adtool -a remove-from-group --user TestUser --fromgroup=TestGroup
```

Remove TestGroup2 from TestGroup:

```
adtool -a remove-from-group --group TestGroup2 --fromgroup=TestGroup
```

Remove TestComputer3 from TestGroup:

```
adtool -a remove-from-group --computer TestComputer3 --fromgroup=TestGroup
```

reset-user-password

Reset a user password.

Reset a user's password reading the password from the TestUser.pwd file:

```
cat TestUser.pwd | adtool -a reset-user-password --name=TestUser --password=- --no-password-
expires
```

search-computer

Search for computer objects, print DNs.

```
/opt/pbis/bin/adtool -d domain.com -a search-computer --search-base OU=Test,DC=domain,DC=com --
scope subtree --name tst-QAmachine
```

search-group

Search for group objects, print DNs.

```
/opt/pbis/bin/adtool -a search-group --search-base=OU=Test,DC=schnauzers,DC=com --scope=subtree -
-name=testgroup0
```

search-object

Search for any type of objects using LDAP filter.

Look up all attributes of an AD object using filter-based search:

```
adtool -a search-object --filter '(&(objectClass=person) (displayName=TestUser))' -t | adtool -a
lookup-object
```

search-ou

Search for organizational units, print DNs.

Look up the description attribute of an OU specified by name with a wildcard:

```
adtool -a search-ou --name='*RootOu' -t | adtool -a lookup-object --dn=- --attrr= description
```

search-user

Search for users, print DNs.

Look up the unixHomeDirectory attribute of a user with sAMAccountName TestUser:

```
adtool -a search-user --name TestUser -t | adtool -a lookup-object --dn=- --attrr=
unixHomeDirectory
```

Look up the userAccountControl attribute of a user with CN TestUserCN:

```
adtool -a search-user --name CN=TestUserCN -t | adtool -a lookupobject --dn=- --
attr=userAccountControl
```

Look up the userAccountControl attribute of a user with CN TestUserCN:

```
adtool -a search-user --name CN=TestUserCN -t | adtool -a lookupobject --dn=- --
attr=userAccountControl
```

set-attr

Set or clear a value for an attribute.

> ***Note:*** *Multi-value entries are limited to 100 entries.*

To set:

```
adtool -a set-attr --dn CN=$HOSTNAME-u,OU=$HOSTNAME-ou,OU=adtool,OU=automation --attrName gecos -
-attrValue "setattr"
```

To clear:

```
adtool -a set-attr --dn CN=$HOSTNAME-u,OU=$HOSTNAME-ou,OU=adtool,OU=automation --attrName gecos
```

To set multi-value:

```
adtool -a set-attr --dn CN=$HOSTNAME-u,OU=$HOSTNAME-ou,OU=adtool,OU=automation --attrName
businessCategory --attrValue "Engineering;QA;Development"
```

To clear multi-value:

```
adtool -a set-attr --dn CN=$HOSTNAME-u,OU=$HOSTNAME-ou,OU=adtool,OU=automation --attrName
businessCategory
```

AD Bridge Cell Management Commands

add-to-cell

Add a user or group to an AD Bridge Cell.

Add group TestGroup to an AD Bridge Cell in TestOU:

```
adtool -a add-to-cell --dn OU=TestOU, DC=department,DC=company,DC=com --group=TestGroup
```

delete-cell

Delete an AD Bridge Cell.

Change the default login shell property of an AD Bridge Cell in TestOU:

```
/opt/pbis/bin/adtool -a delete-cell --dn=OU=Test,DC=domain,DC=com --force
```

edit-cell

Modify AD Bridge Cell properties.

```
/opt/pbis/bin/adtool -a edit-cell --dn=OU=Test,DC=domain,DC=com --default-login-shell=/bin/ksh
```

edit-cell-group

Modify properties of a cell's group.

Change login shell property of TestUser in a cell created in TestOU:

```
adtool -a edit-cell-user --dn OU=TestOU --user TestUser --login-shell=/usr/bin/ksh
```

edit-cell-user

Modify properties of a cell's user.

```
/opt/pbis/bin/adtool -a edit-cell-user --dn=OU=Test,DC=domain,DC=com --
user=CN=testuser,OU=Test,DC=domain,DC=com --uid=123456789
```

link-cell

Link AD Bridge Cells.

Link cell in OU=TestOU1 to the default cell in DC=country:

```
adtool -a link-cell --source-dn OU=TestOU1,DC=department,DC=company,DC=com --target-dn
DC=country,DC=company,DC=com
```

lookup-cell

Retrieve AD Bridge Cell properties.

Find cells linked to an AD Bridge Cell in OU=TestOU,DC=department,DC=company,DC=com:

```
adtool -a lookup-cell --dn OU=TestOU --linked-cells
```

lookup-cell-group

Retrieve AD Bridge Cell properties.

Find cells linked to an AD Bridge Cell in OU=TestOU,DC=department,DC=company,DC=com:

```
adtool -a lookup-cell --dn OU=TestOU --linked-cells
```

lookup-cell-user

Retrieve properties of a cell's user.

Look up login shell property of TestUser in a cell created in TestOU:

```
adtool -a lookup-cell-user --dn OU=TestOU --user TestUser --login-shell
```

new-cell

Create a new AD Bridge Cell.

```
/opt/pbis/bin/adtool -a new-cell --dn=OU=Husky,DC=domain,DC=com --default-login-shell=/bin/bash
```

remove-from-cell

Remove a user or group from a AD Bridge Cell.

Remove TestUser from an AD Bridge Cell in TestOU:

```
adtool -a remove-from-cell --dn OU=TestOU,DC=department,DC=company,DC=com --user=TestUser
```

search-cells

Search for AD Bridge Cells.

Search for cells in a specific location:

```
adtool -a search-cells --search-base OU=department,DC=country,DC=company,DC=com
```

unlink-cell

Unlink AD Bridge Cells.

Unlink cell in OU=TestOU1 from the default cell in DC=country:

```
adtool -a unlink-cell --source-dn OU=TestOU1,DC=department,DC=company,DC=com --target-dn
DC=country,DC=company,DC=com
```

Example: *This example shows how to use two authentication methods and how to search Active Directory even though the computer on which the command was executed was not connected to the domain. The account specified in the options is an Active Directory administrative account.*

```
root@ubuntu:/opt/pbis/bin# ./adtool -a search-cells --search-base dc=connecticut,dc=com -
-logon-as=Administrator --passwd=-
```

In this case, the successful result would be:

```
Enter password:
CN=$LikewiseIdentityCell,DC=connecticut,DC=com
CN=$LikewiseIdentityCell,OU=mySecureOU,DC=connecticut,DC=com
Total cells: 2
```

Additional Commands and Options

To get information about the options for each action, use the following syntax:

```
/opt/pbis/bin/adtool --help -a <ACTION>
```

Example: *Here is an example with the information that is returned:*

```
/opt/pbis/bin/adtool --help -a new-user
Usage: adtool [OPTIONS] (-a |--action) new-user <ARGUMENTS>new-user - create a new user
account.
Acceptable arguments ([X] - required):
--dn=STRING         DN/RDN of the parent container/OU containing the user. (use '-'
                for stdin input)
--cn=STRING         Common name (CN) of the new user. (use '-' for stdin input)
--logon-name=STRING     Logon name of the new user.  (use '-' for stdin input) [X]
--pre-win-2000-name=STRING    Pre Windows-2000 logon name
--first-name=STRING     First name of the new user.
--last-name=STRING      Last name of the new user.
--description=STRING    Description of the user.
--password=STRING       User's password. (use '-' for stdin input)
--no-must-change-password    User is not required to change the password at next logon.
If
                omitted user must change password at next logon unless
                "--no-password-expires' option is specified
--no-password-expires        The password never expires. If omitted - user must change
password
                on next logon.
--account-enabled       User account will be enabled. By default it is disabled on
creation
```

Options

To view the tool's options and to see examples of how to use them, execute the following command:

```
/opt/pbis/bin/adtool --help
```

Example: *Here is an example with the information that is returned:*

```
[root@rhel5d bin]# ./adtool --help Usage: adtool [OPTIONS] <ACTION> [ACTION_ARGUMENTS]

HELP OPTIONS
-u, --usage            Display brief usage message
-?, --help            Show this message, help on all actions (-a), or help on a
                specific action (-a <ACTION>).
-v, --version            Print program version and exit.
```

```
COMMON OPTIONS
-l, --log-level=LOG_LEVEL    Acceptable values: 1 (error), 2(warning), 3(info), 4
(verbose)
                5 (trace) (Default: warning).
-q, --quiet            Suppress printing to stdout. Just set the return code. print-dn
                option makes an exception.
-t, --print-dn            Print DNs of the objects to be looked up, modified or searched
for.
-r, --read-only        Do not actually modify directory objects when executing actions.

CONNECTION OPTIONS
-s, --server=STRING        Active Directory server to connect to.
-d, --domain=STRING        Domain to connect to.
-p, --port=INT            TCP port number
-m, --non-schema         Turn off schema mode

AUTHENTICATION OPTIONS
-n, --logon-as=STRING        User name or UPN.
-x, --passwd=STRING        Password for authentication. (use '-' for stdin input)
-k, --keytab=STRING        Full path of keytab file, e.g. /etc/krb5.keytab
-c, --krb5cc=STRING         Full path of krb5 ticket cache file,
                e.g. /tmp/krb5cc_foo@example.com
-z, --no-sec            Turns off secure authentication. Simple bind will be used. Use
                with caution!

ACTION
-a, --action[=<ACTION>]     Action to execute. Type '--help -a' for a list of actions, or
                '--help -a <ACTION>' for information on a specific action.
Try '--help -a' for a list of actions.
```

Use adtool

Privileges: The **adtool** provides similar features as native Microsoft Active Directory tools. When using **adtool**, be sure to use an account that has appropriate permissions in place to apply changes to Active Directory objects.

For example, to add a user to a security group, you must be a member of a security group, such as the Enterprise Administrators security group.

For more information on Active Directory privileges, permissions, and security groups, see the following references on the Microsoft TechNet website:

- *Active Directory Privileges*
- *Active Directory object permissions*
- *Active Directory Users, Computers, and Groups*
- *Securing Active Directory Administrative Groups and Accounts*

Options: There are short and long options. Separate arguments from options with either space or equal sign. If you are not sure about the results of an action you want to execute, run it in read-only mode first (**-r**). It can also be useful to set the log level to TRACE (**-l 5**) to see all execution steps the tool is taking.

Authentication: The **adtool** uses single sign-on by default if the computer is domain-joined. Otherwise, it uses krb5 using a cached ticket, keytab file, or username and password (unless secure authentication is turned-off (**--no-sec**).

Name resolution: In most cases, you can reference objects by FQDN, RDN, UPN, or names that make sense for a specific action. Use a dash if you want the tool to read values from **stdin**. This allows you to combine commands using pipes, such as search and lookup actions.

Multi-forest support: You can reference an object from a name context (forest) different from the one you are currently connected to, provided that there is a proper trust relation between them. In this way, for instance, you can add a user from one forest to a cell defined in another forest.

Create a New Cell: When you create a new cell, the tool adds the default primary group (domain users) to the cell. If you add a user to the cell and the user has a primary group different from the default group, which is an atypical case, you must also add the primary group to the cell. The tool does not do this automatically.

Add Users or Groups Across Domains: When you add a user or group to a cell, and if the user or group is in a domain different from the one hosting the cell, you must use an account that has write permissions in the cell domain and at least read permissions in the domain hosting the user or group.

For example, you want to add a user such as **CORP\kathy**, whose primary group is **domain users**, to a cell in a domain named **CORPQA**. Two conditions must be met:

- You must be authenticated to the **CORPQA** domain as a user with administrative rights in the **CORPQA** domain.
- Your user account must exist in the **CORP** domain with at least read permissions for the **CORP** domain.

Since, in this example, the primary group of **CORP\kathy** is **CORP\domain users**, you must also add **CORP\domain users** to the cell in the **CORPQA** domain.

Automate Commands with a Service Account: To run the tool under a service account, such as a cron job, avoid using krb5 tickets for authentication, especially those cached by the AD Bridge Enterprise authentication service in the directory. The tickets may expire, and the tool will not renew them. Instead, we recommend that you create an entry for the service account in a keytab file and use the keytab file for authentication.

Work with a Default Cell: The tool uses the default cell only when the value of the parameter is the root naming context, such as when you use an expression like **--dn DC=corp,DC=example,DC=com** to represent **corp.example.com**.

Configure Sudoers File in AD Bridge

Configure Entries in Your sudoers Files

When you add Active Directory entries to your **sudoers** file, typically **/etc/sudoers**, you must adhere to at least the following rules:

- ALL must be in uppercase letters.
- Use a slash character to escape the slash that separates the Active Directory domain from the user or group name.
- Use the correct case; entries are case sensitive.
- Use a user or group alias if the user or group has one in Active Directory.
- If the user or group does not have an alias, you must set the user or group in the AD Bridge Enterprise canonical name format of **NetBIOSdomainName\sAMAccountName** (and escape the slash character).

For users or groups with an alias, the AD Bridge Enterprise canonical name format is the alias, which you must use. You cannot use the format of **NetBIOS domain name\SAM account name**.

For users and groups without an alias, the form of an entry in the **sudoers** file is as follows:

```
DOMAIN\\username
DOMAIN\\groupname
```

Example: *Example entry of a group.*

```
% EXAMPLE\\LinuxFullAdmins ALL=(ALL) ALL
```

Example entry of a user with an alias.

```
kyle ALL=(ALL) ALL
```

*For more information about how to format your **sudoers** file, please see your computer's **man** page for sudo.*

Configure User-Ignore and Group-Ignore

User-Ignore

```
/etc/pbis/user-ignore
```

The file contains a list of local users, one per line, to ignore when authenticating users with AD Bridge. This is used when a local and Active Directory user share the same login name, and you want to ensure the local user has preference when logging in.

By default, the root user is listed in the file. Do not remove the default value.

Edit the file locally or update the group policy setting **Lsass: User names to ignore**. Updates to the file do not require the restart of any AD Bridge services.

Group-Ignore

```
/etc/pbis/group-ignore
```

The file contains a list of local groups, one per line, to ignore when authenticating groups with ADB. This is used when a local and Active Directory group share the same login name, and you want to ensure the local group has preference when logging in.

By default, root and tty groups are listed in the file. Do not remove the default values.

Edit the file locally or update the group policy setting **Lsass: Group names to ignore**. Updates to the file do not require the restart of any AD Bridge services.

Kerberos Commands in AD Bridge

AD Bridge includes several command-line utilities for working with Kerberos. We recommend that you use these Kerberos utilities, located in **/opt/pbis/bin**, to manage those aspects of Kerberos authentication that are associated with AD Bridge Enterprise.

For complete instructions on how to use the Kerberos commands, see the **man** page for the command. For example, **man <command name>**.

> *To address Kerberos issues, see Troubleshooting Kerberos Errors at https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc728430(v=ws.10).*

kdestroy

The **kdestroy** utility destroys the user's active Kerberos authorization tickets obtained through AD Bridge. Destroying the user's tickets can help solve login problems.

This command destroys only the tickets in the AD Bridge Kerberos cache of the user account that is used to execute the **kdestroy** command; tickets in other Kerberos caches, including root, are not destroyed. To destroy another user's cache, use the command with its **- c** option.

klist

Lists Kerberos tickets, including the location of the credentials cache, the expiration time of each ticket, and the flags that apply to the tickets.

Because AD Bridge includes its own Kerberos 5 libraries (in **/opt/pbis/lib**), you must use the AD Bridge **klist** command by either changing directories to **/opt/pbis/bin** or including the path in the command.

Example:

```
-sh-3.00$ /opt/pbis/bin/klist
Ticket cache: FILE:/tmp/krb5cc_593495191
Default principal: hoenstiv@EXAMPLE.COM
Valid starting     Expires            Service principal
07/22/08 16:07:23  07/23/08 02:06:39  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 07/23/08 04:07:23
07/22/08 16:06:39  07/23/08 02:06:39  host/rhel4d.EXAMPLE.COM@
        renew until 07/23/08 04:07:23
07/22/08 16:06:39  07/23/08 02:06:39  host/rhel4d.EXAMPLE.COM@EXAMPLE.COM
        renew until 07/23/08 04:07:23
07/22/08 16:06:40  07/23/08 02:06:39  RHEL4D$@EXAMPLE.COM
        renew until 07/23/08 04:07:23
```

kinit

Obtains and caches an initial ticket-granting ticket for a principal.

kpasswd

The **kpasswd** command changes a Kerberos principal's password.

ktutil

Invokes a shell from which you can read, write, or edit entries in a Kerberos keytab.

You can use **ktutil** to add a keytab file to a non-default location.

When you join a domain, AD Bridge initializes a Kerberos keytab by adding the **default_keytab_name** setting to **krb5.conf** and setting it to **/etc/krb5.keytab**. If the keytab file referenced in **krb5.conf** does not exist, the AD Bridge **domain-join** utility changes the setting to **/etc/krb5.conf**.

You can set the keytab file to be in a location that is different from the default. To do so, you must pre-create the keytab file in the location you want and set a symlink to it in **/etc/krb5.keytab**. Then, you must set the **default_keytab_name** in **/etc/krb5.conf** to point to either the symlink or the real file. The result is that the keytab file will already exist and the AD Bridge domain-join utility will not modify its location setting.

The keytab's format does not let you create a keytab file without a keytab, but you can use **ktutil** to manually create one with a **place-holder** entry. When AD Bridge Enterprise adds your computer to the domain, a correct entry will be added to the file.

```
/opt/pbis/bin/ktutil
ktutil:  addent -password -p nonexistent@nonexistent -k 1 -e RC4-HMAC
Password for nonexistent@nonexistent:
ktutil:  wkt /var/OtherPlace/etc/krb5.keytab
ktutil:  quit
```

kvno

Acquires a service ticket for the specified Kerberos principals and prints out the key version numbers of each.

Certificates Auto Enrollment in AD Bridge

You can manage the auto-enrollment of certificates using the config tool. The following commands can be used to manage certificates and auto enrollment. For more information about a command, run the command with **--detail**.

Example:

```
/opt/pbis/bin/config --detail EnableAutoEnroll
```

For information about managing auto enrollment using group polices, please see the AD Bridge Group Policy Reference Guide at www.beyondtrust.com/docs/ad-bridge/how-to/group-policy.

Authentication

Name of certificate or passphrase.

Example:

```
/opt/pbis/bin/config Authentication " "
```

AutoEnrollPollInterval

Sets the number of seconds that pass before the computer queries the CA service. The interval value is in seconds. Accepted interval values are between 300 and 65535 seconds. The default value is **28800** seconds (8 hours).

Example:

```
/opt/pbis/bin/config AutoEnrollPollInterval 300
```

CertificateTemplateNames

List of certificate template names to auto enroll.

Example:

```
/opt/pbis/bin/config CertificateTemplateNames
```

DeleteCertificatesWhenRemoved

Deletes enrolled certificates when the certificate is removed from the **CertificateTemplateNames** list. Accepted values are **true** and **false**.

Example:

```
/opt/pbis/bin/config DeleteCertificatesWhenRemoved
```

EnableAutoEnroll

Turns on the auto enroll service.

Example:

```
/opt/pbis/bin/config EnableAutoEnroll true
```

EnableWireless

Configures and enables the wireless interface. Accepted values are **true** and **false**.

Example:

```
/opt/pbis/bin/config EnableWireless false
```

EncryptPrivateKey

Certificate enrollment generates a private key file which by default is encrypted. Accepted values are **true** and **false**.

Example:

```
/opt/pbis/bin/config EncryptPrivateKey
```

ManagedCertificateLifecycle

Renews, updates, and removes certificates. Accepted values are **true** and **false**.

Example:

```
/opt/pbis/bin/config ManagedCertificateLifecycle false
```

SecurityType

The security method used for the wireless point.

0: None

1: WPA2-Enterprise

2: WPA2-Personal

Example:

```
/opt/pbis/bin/config SecurityType 1
```

SSID

SSID of wireless router.

Example:

```
root@tst-ubu1404-64:/home/testuser# /opt/pbis/bin/config SSID " "
```

Open a Support Case With BeyondTrust Technical Support

BeyondTrust provides an online knowledge base, as well as telephone and web-based support.

For BeyondTrust Technical Support contact information, please visit www.beyondtrust.com/support.

Before Contacting BeyondTrust Technical Support

To expedite support, collect the following information to provide to BeyondTrust Technical Support:

- AD Bridge Enterprise version: available in the AD Bridge Enterprise Console by clicking **Help > About** on the menu bar
- AD Bridge Enterprise Agent version and build number
- Linux or Unix version
- Windows or Windows Server version

If you are contacting BeyondTrust Technical Support about one of the following problems, also provide the diagnostic information specified.

Segmentation Faults

Provide the following information when contacting BeyondTrust Technical Support:

- Core dump of the AD Bridge application:

  ```
  ulimit - c unlimited
  ```

- Exact patch level or exact versions of all installed packages

Program Freezes

Provide the following information when contacting BeyondTrust Technical Support:

- Debug logs
- tcpdump
- An **strace** of the program

Domain-Join Errors

Provide the following information when contacting BeyondTrust Technical Support:

- Debug logs: copy the log file from **/var/log/pbis-join.log**
- tcpdump

All Active Directory Users Are Missing

Provide the following information when contacting BeyondTrust Technical Support:

- Run **/opt/pbis/bin/get-status**
- Contents of **nsswitch.conf**

All Active Directory Users Cannot Log On

Provide the following information when contacting BeyondTrust Technical Support:

- Output of **id <user>**
- Output of **su -c 'su <user>' <user>**
- **lsass** debug logs

> *For more information, please see Generate Debug Logs in the AD Bridge Troubleshooting Guide at www.beyondtrust.com/docs/ad-bridge/how-to/troubleshoot.*

- Contents of **pam.d/pam.conf**
- The sshd and ssh debug logs and syslog

AD Users or Groups are Missing

Provide the following information when contacting BeyondTrust Technical Support:

- The debug logs for lsass
- Output for **getent passwd** or **getent group** for the missing object
- Output for **id <user>** if user
- tcpdump
- Copy of lsass cache file.

Poor Performance When Logging On or Looking Up Users

Provide the following information when contacting BeyondTrust Technical Support:

- Output of **id <user>**
- The lsass debug log
- Copy of lsass cache file.

> *For more information about the file name and location of the cache files, please see the AD Bridge Linux Administration Guide at www.beyondtrust.com/docs/ad-bridge/getting-started/linux-admin.*

- tcpdump

Generate a Support Pack

The AD Bridge support script will copy system files that AD Bridge needs to function into an archive. This archive can then be sent to support to assist in the investigation.

Installed location:

/opt/pbis/libexec/pbis-support.pl