Active Directory GPO support

Canonical Group Ltd

Nov 23, 2023
CONTENTS

1 Tutorials 3
  1.1 Certificates Auto-Enrollment ................................................................. 3
  1.2 Certificates Auto-Enrollment ................................................................. 4

2 How-to guides 5
  2.1 Installation ................................................................................................. 5
  2.2 Operations ................................................................................................. 14

3 Technical Reference 29
  3.1 The adsysctl command .............................................................................. 29
  3.2 The adsys daemon ..................................................................................... 35
  3.3 Active Directory Watch Daemon ............................................................... 40
  3.4 adsysctl command line ............................................................................ 40
  3.5 adsysd command line .............................................................................. 53
  3.6 adwatchd command line ......................................................................... 61
  3.7 Policies reference list ............................................................................... 70
  3.8 Supported releases .................................................................................... 111
  3.9 Recommended readings ............................................................................ 111

4 Explanation 113
  4.1 Managers .................................................................................................. 113

5 In this documentation 145

6 Project and community 147
ADSys is the Active Directory Group Policy client for Ubuntu.

ADSys enables system administrators to manage Ubuntu Desktop clients centrally via Microsoft Active Directory. It includes ADMX and ADML templates and allows policies to be applied at boot and login, then refreshed periodically. It simplifies the task of configuring Ubuntu systems in a Microsoft Active Directory environment. However, it doesn't handle user authentication or initial security policy, which are managed by SSSD/Winbind and PAM.

ADSys is valuable for system administrators who wish to manage Ubuntu Desktop clients in a centralized manner through Microsoft Active Directory.
This section contains step-by-step tutorials to help you get started with ADSys. We hope our tutorials make as few assumptions as possible and are accessible to anyone with an interest in ADSys. They should be a great place to start learning about ADSys, how it works, and what it’s capable of.

1.1 Certificates Auto-Enrollment

*Certificate Auto-Enrollment* is a key component of Ubuntu’s Active Directory GPO support. This feature enables clients to seamlessly enroll for certificates from Active Directory Certificate Services. In this demonstration, we’ll guide you through the entire process, starting from the initial setup requirements to configuring the Active Directory policy. This demonstration is designed to provide you with a hands-on understanding of how to efficiently implement and manage certificate auto-enrollment, ensuring your systems remain secure and compliant with organizational policies.
1.2 Certificates Auto-Enrollment

Explore the essentials of Ubuntu’s Certificate Auto-Enrollment Feature, a crucial tool for seamless certificate management with Active Directory Certificate Services.

- Certificates auto-enrollment demo
These guides accompany you through the complete ADSys operations lifecycle.

2.1 Installation

2.1.1 Windows Domain Controller

This is the configuration needed on Windows Domain Controller.

How to set up the Active Directory Server

Active Directory needs policy files (.admx and .adml) to know which settings can be changed on given clients and actions to be performed on users of the domain.

As a rule of thumb, we generally separate the Ubuntu configuration from the Windows one to avoid incompatibilities:

- namespace conflicts
- forward vs backward slashes
- let the administrator know exactly what settings are supported on which client version.

Ubuntu administrative template generations

ADSys ships with pre-built Active Directory administrative templates that you can install on your Active Directory server. You will find two flavors of them:

- One listing only Long Term Supported (LTS) Ubuntu versions.
- One listing all currently supported Ubuntu versions, including non LTS.

To get them, run `adsysctl policy admx lts-only`. It will create 2 files `Ubuntu.adml` and `Ubuntu.admx` in the current directory. As described, those will list only LTS versions of Ubuntu. To get all supported releases, replace `lts-only` with `all`.

Those are the 2 files that must be copied on your Active Directory server.

You can find the latest version of those files in a dedicated directory of the upstream repository. Note though that not all the keys may be supported by our local ADSys installation. Only the templates generated by `adsysctl` match the version of your client.

The policy files are also shipped as part of the `adsys-windows` package, together with the `Active Directory Watch Daemon`. 
Deployment of ADM files on the Active Directory server

The administrative templates for Ubuntu must be deployed on your Active Directory server in the policy definition directory corresponding to your forest root. For instance `\example.com\sysvol\example.com\Policies\PolicyDefinitions` for the `.admx` file and `\example.com\sysvol\example.com\Policies\PolicyDefinitions\en-US` for the `.adml` file. These directories can be created manually if they do not exist.

For more information read the Microsoft documentation “create and manage the Central Store”.

Once loaded successfully in Active Directory, the Ubuntu specific settings are available in the Group Policy Management Editor under `[Policy Name] > Computer Configuration > Policies > Administrative Templates > Ubuntu` for the machine policies and `[Policy Name] > User Configuration > Policies > Administrative Templates > Ubuntu` for the user policies.

![Group Policy Management Editor](image.png)
Recommended readings

- adsysctl help policy admx or man adsysctl-policy-admx.
- Create and manage the Central Store.
Active Directory GPO support

Active Directory Watch Daemon

The Active Directory Watch Daemon (or adwatchd) is a Windows application geared towards automating the manual process of incrementing the version stanza of a GPT.ini file.

At its core, the program can be simplified to the following steps:

• watch a list of user-configured directories for changes – subdirectories are also watched, but only the root directory will have a GPT.ini file
• when a change is detected, attempt to locate a GPT.ini file at the root of the watched directory, or create one if absent
• if a GPT.ini file is found, increment the version stanza of the file by 1, thus signaling clients that a new version of the assets (including scripts) are available to download during the next client refresh

Installation

The adwatchd executable is available as a standalone Windows executable file, distributed as part of the adsys-windows Ubuntu package, or packaged as an installer available on the GitHub repository.

Installing via the Ubuntu package

To source the adwatchd executable from the adsys-windows Ubuntu package, we must run the following (on Ubuntu):

```
sudo apt install adsys-windows
```

After a successful installation, the adwatchd.exe executable will be available in the /usr/share/adsys/windows directory. We suggest you deploy this executable to a persistent directory of your choosing on the AD Domain Controller, such as %SystemDrive%\Program Files\Ubuntu\adsys\.

Installing via the bespoke installer

Download the latest release of the adwatchd_setup.exe file (or a specific version if you wish), and run it.

Follow the installation steps, paying attention to the prompts, optionally specifying a different installation directory for the application.

Configuring and starting the daemon

Regardless of how the application is installed, the configuration steps are the same:

• locate and run the adwatchd.exe executable to start the application’s interactive configuration tool
• specify a path for the configuration file, or leave it blank to use the default location (the directory where the executable is located)
• specify a list of directories to watch, one per line (the program will block installation if any of the directories do not exist)
• hit the [ Install ] button to finish the installation, this will:
  – create the configuration file if it does not exist
  – install and start the adwatchd Windows service
Active Directory GPO support

For a better understanding on what directories should be configured for watching, please refer to the Installing scripts on sysvol section of the [Scripts execution](/explanation/scripts#Installing scripts on sysvol) document.

Note that the interactive configuration tool can only be run if the adwatchd service is not already installed on the machine. Please refer to the [CLI usage](#CLI usage) section for instructions on how to finely manage the service.

We recommend making use of the interactive configuration tool to install the application, as it provides a level of error handling, taking care of path normalization and the creation of the configuration file.

The configuration file is stored as a YAML file, and can be freely edited after the application has been installed. The following keys are configurable:

```
verbose: 0  # 0 = warning, 1 = info, 2 = debug, 3 = debug with caller output
dirs:       # list of directories to watch
  - C:\Windows\SYSVOL\sysvol\testdomain.com\Ubuntu  # traditional path
  - \testdomain.com\SYSVOL\testdomain.com\Ubuntu    # UNC path
```

Configuring the service using a pre-filled configuration file

For convenience, the adwatchd application can be configured with a pre-filled configuration file. Start a Command Prompt or PowerShell window and run one of the following:

```
REM Run the interactive configuration tool with a predefined configuration file
REM This will start the interactive configuration tool with pre-filled entries for the config path and directories to watch, leaving the user to press the [ Install ] button
C:\path\to\adwatchd.exe -c path\to\config.yaml

REM Run the service installation command with a predefined configuration file
REM This will install the service with the given configuration file and start it
C:\path\to\adwatchd.exe service install -c path\to\config.yaml
```

Upgrading

The upgrade process differs based on the installation method used. If you decide to switch to a different installation method, you will need to uninstall the existing service beforehand.

Upgrading via the Ubuntu package

1. Source the new adwatchd executable from the adsys-windows Ubuntu package
2. Stop the adwatchd service (via the Services GUI or the adwatchd service stop command)
3. (Optional) Remove the existing adwatchd service from the system (through the adwatchd service uninstall command)
4. Replace the existing adwatchd.exe executable with the new one
5. (Optional) Install the adwatchd service (through the adwatchd service install command)
6. Start the adwatchd service (via the Services GUI or the adwatchd service start command)
Active Directory GPO support

The optional steps are only necessary if the intent is to do a complete upgrade of the application and are not usually needed. Always refer to the changelog for information on the latest version of the application.

Upgrading via the bespoke installer

1. Source the latest release of the adwatchd_setup.exe file
2. Run the installer, following the prompts
The installer will automatically take care of the upgrade process, and will offer to stop the service if it is running prior to the upgrade, and start it afterwards.

2.1.2 Linux client machine

This section outlines how to join your client machine to the domain, install ADSys, and verify that the setup is correct.

How to join an Active Directory domain during installation

In order to use Group Policies on an Ubuntu client, the first thing to do is, of course, to join the computer to an Active Directory domain.

A machine can join an AD domain at installation time with the Ubuntu Desktop installer, or after installation, by manually setting up the connection to AD.

Join at installation time

Joining during installation is only supported by the Ubuntu Desktop graphical installer Ubiquity. So, start an installation of Ubuntu Desktop as you would usually do and proceed to the page “Who are you?” Enter user and computer name information.
Note about the host name:

In order to set and resolve the host name properly, you must enter the **Fully Qualified Domain Name** (FQDN) of the machine in the field “Your computer’s name”. For example, `host01.example.com` instead of only the host name `host01`.

After installation you can check if it is correct with the command `hostname` and `hostname -f` which must return the name of the machine (`host01`) and the full name of the machine with the domain (`host01.example.com`) respectively.

Check the box “Use Active Directory” and click “Continue” to proceed with next step “Configure Active Directory”.

On this page you can enter the address of the Active Directory controller and credentials of the user allowed to add machines to the domain.
You can verify that the server is reachable by pressing “Test Connection”.

Once all the information has been entered and is valid, press “Continue” to proceed with the remaining usual steps of the installation.

At the end of the installation you can reboot the machine and you are ready to log in as a user of the domain on first boot.

If anything goes wrong with the join process during installation, you will be notified by a dialog box. You can still reboot the machine, log in as the administrator user of the machine (i.e. the user you entered in the page “Who are you?”) and troubleshoot the issue. The Ubuntu Server Guide provides instructions to perform such troubleshooting.

**How to join an Active Directory domain manually**

ADSys supports two Active Directory backends:

1. **SSSD**, or System Security Services Daemon, provides access to centralized identity management systems like Microsoft Active Directory, OpenLDAP, and various other directory servers. This client component retrieves and caches data from remote directory servers, delivering identity, authentication, and authorization services to the host machine.

2. **Winbind** is a component of the Samba suite that provides seamless integration and authentication services between UNIX or Linux systems and Windows-based networks, allowing the former to appear as members in a Windows Active Directory domain.
Active Directory GPO support

Join manually using SSSD

The aim of this documentation is to describe how to operate ADSys. So we won’t do an in depth description of the operations to manually configure a connection to Active Directory from an Ubuntu Client.

Authentication of Ubuntu against the Active Directory server requires to configure SSSD and Kerberos. SSSD will then retrieve the credentials and the initial security policy of the Default Domain Policy.

All these operations are described in details in the Ubuntu Server Guide “Service - SSSD” and the White Paper How to integrate Ubuntu Desktop with Active Directory.

Join manually using Winbind

In addition to SSSD, ADSys supports Winbind as a backend. The easiest way to join a domain using Winbind is to use the realmd utility, as described in the Samba - Active Directory guide.

ADSys uses SSSD as a default backend, so Winbind has to be opted into explicitly via the following configuration option in adsys.yaml:

```
ad_backend: winbind
```

In addition, Winbind requires additional dependencies to be installed. On Ubuntu-based systems they can be installed by executing the following command, prior to ADSys:

```
sudo apt update
sudo apt install winbind krb5-user
```

How to set-up ADSys

ADSys is not currently installed by default on Ubuntu desktop. This must be done manually by the local administrator of the machine.

To do so, log in on first boot, update the repositories and install ADSys. On Ubuntu-based systems this can be accomplished with the following commands:

```
sudo apt update
sudo apt install adsys
```

Reboot then to allow the machine to do its policy refresh.

Logging in as a user of the domain

To log in as a user of the domain, press the link “Not listed?” in the greeter. Then enter the username followed by the password.
SSSD

By default, there is no default domain configured in SSSD. You have to enter the full user name with one of the forms: USER@DOMAIN.COM, USER@DOMAIN or DOMAIN/USER.

On the first log in the user’s home directory is created.

All of this (default domain, default path for home directories, default shell, etc.) is configurable in /etc/sssd/sssd.conf.

Winbind

If Winbind is used as a backend, the account can be specified in one of the following forms: USER@DOMAIN.COM, USER@DOMAIN or DOMAIN\USER.

For the home directory to be created automatically on login, the pam_mkhomedir module can be enabled:

```
sudo pam-auth-update --enable mkhomedir
```

Options such as the home directory path template, shell and others can be tweaked in /etc/samba/smb.conf and are documented in the smb.conf(5) man page.

2.2 Operations

2.2.1 How to use GPO with Ubuntu

As explained in previous chapter, there are 2 sets of Ubuntu specific settings in the Group Policy Management Editor:

- [Policy Name] > Computer Configuration > Policies > Administrative Templates > Ubuntu for the machine policies.
- [Policy Name] > User Configuration > Policies > Administrative Templates > Ubuntu for the user policies.

Your first Ubuntu GPO rule

For this example we will use a test domain called warthogs.biz with 2 separate OUs.

- The machine is called adclient04 and belongs to warthogs.biz > MainOffice
• The user is called bob and belongs to warthogs.biz > IT Dept > RnD
In this example, we will demonstrate how to change dconf settings. We will first modify the greeter background image to illustrate how to enforce a computer setting and the list of preferred applications in the launcher for the user settings.

### Modifying a computer setting

Launch the GPO Management editor and create a GPO in warthogs.biz > MainOffice

1. Select GDM background picture setting in Computer Configuration > Policies > Administrative Templates > Ubuntu > Login Screen > Interface > Picture URI.
2. Select Enabled to enable the modification of the Picture URI field.
3. Enter a valid absolute path to a .png image on the client machine, e.g. /usr/share/backgrounds/ubuntu-default-greyscale-wallpaper.png.
4. Refresh the GPO rule on the client by rebooting the machine or running adsysctl update -m (You may be prompted to enter your password to check if have enough privileges to run the command)
The change is now visible on the greeter.
Files are not copied by the Active Directory client and must already exist on the target system at this given path.

Modifying an user setting

1. Let’s create another GPO in warthogs.biz > IT Dept > RnD.
2. Select the list of favorite desktop applications setting in User Configuration > Policies > Administrative Templates > Ubuntu > Desktop > Shell > List of desktop file IDs for favorite applications.
3. Enter a list of valid .desktop file IDs, one per line, like the following:

```
libreoffice-writer.desktop
snap-store_ubuntu-software.desktop
yelp.desktop
```
4. Refresh the GPO rule applied to the user by logging in or running `adsysctl update` as your current user or `adsysctl update --all` to refresh the computer and all active users.

The list of applications showing up on the left side for your current Active Directory user should be updated.
There are other ways of defining a list in ADSys / Active Directory integration. Check the “Different types of widgets” section below.

**General principles**

There are multiple **policy managers** for different types of settings. As of now, only a **dconf** manager is available. When settings are common to a machine and users, the machine settings will always take precedence over the user ones.

The workflow to update a setting in the **GPO Management editor** and to apply the setting to a target user or machine is similar to Windows clients. However, we will see below that there are slight differences when the GPO are applied and refreshed between Windows and Ubuntu.

**When are GPO applied?**

Any change to a GPO is applied:

- On boot for the machine settings
- On login for the user settings
- A periodic refresh timer will update the GPOs of the machine and all active users.

Next section will detail how to configure this and what happens when the Active Directory controller is unreachable.
State of GPO settings

Most GPO rules can have 3 states: enabled, disabled, not configured. These states may have different meanings depending on the manager.

General information of a setting

The left pane of the GPO Management Editor contains the options that can be edited when a setting is enabled.

There is a default value for all the releases and an override for each supported release of Ubuntu. More about multiple releases in the next section.
The **right pane** of the GPO Management editor contains the general information about the GPO including:

- The description of the setting
- The type of settings (e.g. dconf)
- The path of the key in our schema
- The default value of the key that is used if nothing is set on the left pane. Note that if defaults differ between releases, this will be a list per release.
- The list of releases that support this setting.

**Different types of widgets**

**Text entry**

The type **Text** represents a single line of text. If you don't enclose a string with single quotes `'` and the value is not a decimal, it will be done automatically and the entry will be sanitized (e.g. space, `...'`). If you want to force a decimal to be treated as a string, enclose the value with single quotes.

The default value will be already set.
Text list

A multiline text field is used for this case. A list can be:

- One item per line: any end of line will be considered as a delimiter Example:
Active Directory GPO support

libreoffice.desktop
firefox.desktop
nautilus.desktop

• Multiple items on one line: a comma , is the item delimiter:

libreoffice.desktop, firefox.desktop, nautilus.desktop

Note that spaces will be stripped automatically.

• Both syntaxes can be combined:

libreoffice.desktop, firefox.desktop
nautilus.desktop

The type can be either text or numeric:

• Text list:

libreoffice.desktop
firefox.desktop
nautilus.desktop

• Decimal list:

42
300
10

Ensure that you enter the valid type of list, as expected by dconf setting. ADSys will do its best to try to match the entry with the right and expected dconf type.

String or decimal field?

The type of the value will be detected automatically and interpreted as a number if it contains only digits and no quotes, everything else is considered a string. If you want to enforce a list to have string entries, enclose each entry with single quotes.

Text list:

’42’
’300’
’10’

Dropdown list

A list field is a limited list of values to choose from. It is represented by a drop down list.
A checkbox will correspond to set to true or false values for the corresponding setting. The default value will be already selected.
Active Directory GPO support

Decimal

Decimal values are fields that allow only digits with optional upper and lower bounds. A spinner helps the user to increase or decrease the value.

The limits, if any, will be specified in the right section, per release.

Multi-release support

ADSys supports setting different values for different releases of Ubuntu.

The top entry will set a common value between all clients, independently of the release it's running on.

If you need to specify a per-release value for a set of clients, select the Override checkbox for the corresponding release and enter a value in the associated field. If the Override is checked, but no value is specified, the default value for this release will be used.

By definition, override takes precedence over the default value defined at the top for all the releases.

Finally, note that the help text on the right panel will list each default per release if they differ between themselves. In addition, it will list the supported releases for this setting.
Multi-release overrides are only available when your Active Directory administrative templates defines more than one release. If this is not the case, you will only see the top entry to define your policy.
This section consolidates technical details on ADSys, including specifications, APIs, and architecture.

On the Linux side, ADSys is composed of a daemon and a command line interface:

- The daemon - `adsysd` - implements the Group Policy protocol. It relies on Kerberos, Samba and LDAP for authentication and policy retrieval.
- The command line interface - `adsysctl` - controls the daemon and reports its status.

A Windows daemon, `adwatchd` can be installed on the domain controller to automatically refresh assets without system administrator interventions.

### 3.1 The `adsysctl` command

`adsysctl` is a command line utility to request actions from the daemon and query its current status. You can get more verbose output with the `-v` accumulative flags, which will stream all logs from the service corresponding to your specific request.

As a general rule, favor shell completion and the help command for discovering various parts of the `adsysctl` user interface. It will help you by completing subcommands, flags, users and even chapters of the offline documentation!

#### 3.1.1 Which policies are applied

- You can check with policies are currently applied to your current AD user with the `adsysctl policy applied` command:

```bash
$ adsysctl policy applied
Policies from machine configuration:
- MainOffice Policy 2 ({B8D10A86-0B78-4899-9AF-6F0124ECEB48})
- MainOffice Policy ({C4F393CA-AD9A-4595-AEBC-3FA6EE484285})
- Default Domain Policy ({31B2F340-016D-11D2-945F-00C04FB984F9})

Policies from user configuration:
- RnD Policy 3 ({073AA7FC-5C1A-4A12-9AFC-42EC9C5CAF04})
- RnD Policy 2 ({83A5BD5B-1D5D-472D-827F-DE0E6F714300})
- RnD Policy ({5EC4DF8F-4F4E-41DE-846B-52A6FFAF242})
- IT Policy ({75545F76-DEC2-4ADA-B7B8-D5209FD48727})
- Default Domain Policy ({31B2F340-016D-11D2-945F-00C04FB984F9})
```
Active Directory GPO support

The order of policies are top-down, higher GPOs have priorities over lower ones on the stack (respecting OU order, GPO enforcement, GPO block instructions on your AD setup...).

- If you have the right permission, you can request other users as well:

```bash
$ adsysctl policy applied tina
```

Policies from machine configuration:
- MainOffice Policy (\{A2F393CA-AD9A-4595-AEBC-3FA6EE484285\})
- Default Domain Policy (\{31B2F340-016D-11D2-945F-00C04FB984F9\})

Policies from user configuration:
- RnD Policy 4 (\{25A5BD5B-1D5D-472D-827F-DE0E6F714300\})
- IT Policy (\{75545F76-DEC2-4ADA-B7B8-D5209FD48727\})
- Default Domain Policy (\{31B2F340-016D-11D2-945F-00C04FB984F9\})

Pro tip! Use shell completion to get the list of active users you can request which policies are applied on.

- To get which policy are set to a given value or disabled by which key, use the `--details` flags:

```bash
$ adsysctl policy applied --details
```

Policies from machine configuration:
- MainOffice Policy (\{B8D10A86-0B78-4899-91AF-6F0124ECEB48\})
- dconf:
  - org/gnome/desktop/notifications/show-banners: Locked to system default
- MainOffice Policy (\{C4F393CA-AD9A-4595-AEBC-3FA6EE484285\})
- dconf:
  - org/gnome/desktop/notifications/show-banners: Locked to system default
- dconf/org/gnome/desktop/notifications/show-banners: 24h
  - dconf/org/gnome/desktop/notifications/show-weekday: true
  - dconf/org/gnome/desktop/notifications/show-date: false
- dconf/org/gnome/desktop/screensaver/picture-uri: 'file:///usr/share/backgrounds/ubuntu-default-greyscale-wallpaper.png'
- Default Domain Policy (\{31B2F340-016D-11D2-945F-00C04FB984F9\})

Policies from user configuration:
- RnD Policy 4 (\{25A5BD5B-1D5D-472D-827F-DE0E6F714300\})
- dconf:
  - org/gnome/desktop/media-handling/automount: Locked to system default
- RnD Policy 2 (\{83A5BD5B-1D5D-472D-827F-DE0E6F714300\})
- dconf:
  - org/gnome/shell/favorite-apps: libreoffice-writer.desktop
  - snap-store_ubuntu-software.desktop
  - yelp.desktop
- IT Policy (\{75545F76-DEC2-4ADA-B7B8-D5209FD48727\})
- dconf:
  - org/gnome/shell/background/picture-options: stretched
  - org/gnome/shell/background/picture-uri: file:///usr/share/backgrounds/canonical.png
- Default Domain Policy (\{31B2F340-016D-11D2-945F-00C04FB984F9\})

- The `--all` flag will list every key set by a given GPO, including the ones that are redefined by another GPO with a higher priority. This is traditionally helpful for debugging your GPO stack and discover where a given value is defined:

```bash
$ adsysctl policy applied --all
```

Policies from machine configuration:

(continues on next page)
Active Directory GPO support

- MainOffice Policy 2  
  - gdm:
  - dconf/org/gnome/desktop/notifications/show-banners: Locked to system default
- MainOffice Policy
  - gdm:
  - dconf/org/gnome/desktop/interface/clock-format: 24h
  - dconf/org/gnome/desktop/interface/clock-show-date: false
  - dconf/org/gnome/desktop/interface/clock-show-weekday: true
  - dconf/org/gnome/desktop/screensaver/picture-uri: 'file:///usr/share/backgrounds/ubuntu-default-greyscale-wallpaper.png'
- Default Domain Policy

Policies from user configuration:
- RnD Policy 3
  - dconf:
    - org/gnome/desktop/media-handling/automount: Locked to system default
- RnD Policy 2
  - dconf:
    - org/gnome/desktop/media-handling/automount: Locked to system default
- RnD Policy
  - dconf:
    - org/gnome/shell/favorite-apps: libreoffice-writer.desktop
    - dconf
      - snap-store_ubuntu-software.desktop
      - snap-store_ubuntu-software_argusdesktop
      - snap-store_ubuntu-software_yelp.desktop
- IT Policy
  - dconf:
    - org/gnome/desktop/background/picture-options: stretched
    - org/gnome/desktop/background/picture-uri: file:///usr/share/backgrounds/canonical.png
- Default Domain Policy

3.1.2 Refreshing the policies

The command `adsysctl policy update` is used to refresh the policies. By default only the policy of the current user is updated. It can also refresh only the policy of the machine with the flag `-m`, or the machine and all the active users with the flag `-a`. On success nothing is displayed.

For example, refreshing the policy for all the objects:

```
$ adsysctl policy update --all -v
INFO No configuration file: Config File "adsys" Not Found in "/[home/warthogs.biz/b/bob /etc]".
We will only use the defaults, env variables or flags.
INFO Apply policy for adclient04 (machine: true)
INFO Apply policy for bob@warthogs.biz (machine: false)
```

You can provide the name of a user and the path to its Kerberos ticket to refresh a given user.

For example for user bob@warthogs.biz

```
$ adsysctl update bob@warthogs.biz /tmp/krb5cc_1899001102_wB1bck -vv
INFO No configuration file: Config File "adsys" Not Found in "/[home/warthogs.biz/b/bob /etc]".
```
We will only use the defaults, env variables or flags.
DEBUG Connecting as [26812:519495]
DEBUG New request /service/UpdatePolicy
DEBUG Requesting with parameters: IsComputer: false, All: false, Target: bob@warthogs.biz, Krb5Cc: /tmp/Krb5Cc_189900102_cWbLbcK
DEBUG Check if grpc request peer is authorized
DEBUG Polkit call result, authorized: true
DEBUG GetPolicies for "bob@warthogs.biz", type "user"
DEBUG GPO "RnD Policy 3" for "bob@warthogs.biz" available at "smb://warthogs.biz/SysVol/warthogs.biz/Policies/73AA7FC-5C1A-4A12-9AF5-48C95CAFB04"
DEBUG GPO "RnD Policy 2" for "bob@warthogs.biz" available at "smb://warthogs.biz/SysVol/warthogs.biz/Policies/83A5BD5B-1D5D-472D-827F-DE0E6F714300"
DEBUG GPO "RnD Policy" for "bob@warthogs.biz" available at "smb://warthogs.biz/SysVol/warthogs.biz/Policies/5EC4DF8F-FF4E-41DE-846B-529A6FFFAF242"
DEBUG GPO "IT Policy" for "bob@warthogs.biz" available at "smb://warthogs.biz/SysVol/warthogs.biz/Policies/7554F76-DEC2-4A0-7B88-D5209FD4B727"
DEBUG GPO "Default Domain Policy" for "bob@warthogs.biz" available at "smb://warthogs.biz/sysvol/warthogs.biz/Policies/31B2F340-016D-11D2-945F-00C4FB984F9"
DEBUG Analyzing GPO "Default Domain Policy"
DEBUG Analyzing GPO "IT Policy"
DEBUG Analyzing GPO "RnD Policy 2"
DEBUG Analyzing GPO "RnD Policy 3"
DEBUG Analyzing GPO "RnD Policy"
DEBUG Policy "RnD Policy 2" doesn't have any policy for class "user" open /var/cache/adsys/gpo_cache/83A5BD5B-1D5D-472D-827F-DE0E6F714300/User/Registry.pol: no such file or directory
DEBUG Policy "Default Domain Policy" doesn't have any policy for class "user" open /var/cache/adsys/gpo_cache/31B2F340-016D-11D2-945F-00C4FB984F9/User/Registry.pol: no such file or directory
INFO Apply policy for bob@warthogs.biz (machine: false)
DEBUG ApplyPolicy dconf policy to bob@warthogs.biz
DEBUG Update user profile /etc/dconf/profile/bob@warthogs.biz
DEBUG Analyzing entry {Key:org/gnome/desktop/background/picture-options Value:stretched Disabled:false Meta:s}
DEBUG Analyzing entry {Key:org/gnome/desktop/background/picture-uri Value://usr/share/backgrounds/canonical.png Disabled:false Meta:s}
DEBUG Analyzing entry {Key:org/gnome/desktop/media-handling/automount Value: Disabled:true Meta:}
DEBUG Analyzing entry {Key:org/gnome/shell/favorite-apps Value:libreoffice-writer.desktop snap-store_ubuntu-software.desktop yelp.desktop Disabled:false Meta:as}
3.1.3 Getting the status

The status of the service is provided by the command `adsysctl service status`

```
$ adsysctl service status
Machine, updated on Tue May 18 12:15
Connected users:
   bob@warthogs.biz, updated on Tue May 18 12:15

Active Directory:
   Server: ldap://adc01.warthogs.biz
   Domain: warthogs.biz

SSS:
   Configuration: /etc/sssd/sssd.conf
   Cache directory: /var/lib/sss/db

Daemon:
   Timeout after 2m0s
   Listening on: /run/adsysd.sock
   Cache path: /var/cache/adsys
   Run path: /run/adsys
   Dconf path: /etc/dconf
```

You can get the list of connected users, when they were last refreshed, when the next refresh is scheduled and various service configuration options (static or dynamically configured).

3.1.4 Debugging

The `cat` command has already been described in the previous chapter. You can display logs with debugging levels independent of daemon and clients debugging levels. Local printing will also be forwarded.

For example, running `cat` while the command `version` and `applied` are executed:

```
# adsysctl service cat -vv
INFO No configuration file: Config File "adsys" Not Found in "[/root /etc]".
We will only use the defaults, env variables or flags.
DEBUG Connecting as [[29220:823925]]
DEBUG New request /service/Cat
DEBUG Requesting with parameters:
DEBUG Check if grpc request peer is authorized
DEBUG Authorized as being administrator
INFO New connection from client [[29302:462445]]
DEBUG [[29302:462445]] New request /service/Version
DEBUG [[29302:462445]] Requesting with parameters:
DEBUG [[29302:462445]] Check if grpc request peer is authorized
DEBUG [[29302:462445]] Any user always authorized
DEBUG Request /service/Version done
INFO New connection from client [[29455:217212]]
DEBUG [[29455:217212]] New request /service/DumpPolicies
DEBUG [[29455:217212]] Requesting with parameters: Target: bob@warthogs.biz, Details:
   → false, All: false
DEBUG [[29455:217212]] Check if grpc request peer is authorized
```

(continues on next page)
3.1.5 Other commands

Versions

You can get the current service and client versions with the `version` command to check you are running with latest version on both sides:

```
$ adsysctl version
adsysctl 0.5
adsysd 0.5
```

Documentation

An offline version of this documentation is available in the daemon. It will render the documentation on the command line.

You can get a list of all chapters with their titles:

```
$ adysctl doc

  Table of content

  1. [Welcome] ADSys: Active Directory Group Policy integration
  2. [Prerequisites] Prerequisites and installation
  [...]```

And render a given chapter by requesting it:

```
$ adysctl doc Welcome

  ADSys: Active Directory Group Policy integration

  ADSys is the Active Directory Group Policy client for Ubuntu. It allows
  [...]```

Finally, there are different rendering modes to dump documentation in html for instance with the `--format` flag.
Admx generation

The policy admx commands dumps pre-built Active Directory administrative templates that can be deployed on the Active Directory server. For more information, check the AD setup documentation.

Stopping the service

If you do not wish to wait for the idling timeout to stop the server, you can request graceful shutdown with adsysctl service stop. This will first wait for all active connections to ends before shutting down.

The -force flag will end the service immediately.

3.2 The adsys daemon

3.2.1 Policy enforcement

When are the policies refreshed

On the client the policies are refreshed in 3 situations:

- At boot time for the policy of the machine.
- At login time for the policy of the user.
- Periodically by a timer for the machine and the user policy.

What happens when a policy refresh fails

When the client is offline, e.g. a laptop, or the Active Directory server is unreachable, you still want to use the machine and be able to log in. For this purpose, ADSys uses a cache located in /var/cache/adsys.

There are 2 types of cache:

- A cache for the GPO downloaded from the server in directory gpo_cache
- A cache for the rules as applied by ADSys in policies

The enforcement of the policy will fail when the cache is empty or the client fails to retrieve the policy from the server.

If the enforcement of the policy fails:

- At boot time, ADSys stops the boot process.
- At login time, login is denied.
- During periodic refresh, the policy currently applied on the client remains.
How to change refresh rate

Periodic refresh of the policies (machine and active users) is handled by the systemd timer unit `adsys-gpo-refresh.timer`. You can list the timers with the command `systemctl list-timers`.

```
$ systemctl list-timers
NEXT LEFT LAST PASSED
→ UNIT ACTIVATES
Tue 2021-05-18 10:05:49 CEST 11min left Tue 2021-05-18 09:35:49 CEST 18min ago
→ adsys-gpo-refresh.timer adsys-gpo-refresh.service
Tue 2021-05-18 10:31:34 CEST 36min left Tue 2021-05-18 09:31:09 CEST 23min ago
→ anacron.timer anacron.service
[...]
```

The default refresh rate is **30 minutes**.

You can override the refresh rate with the configuration variables `OnBootSec` and `OnUnitActiveSec` in a configuration file as follow:

```
/etc/systemd/system/adsys-gpo-refresh.timer.d/refresh-rate.conf:

# Refresh ADSys GPO every 2 hours
[Timer]
OnBootSec =
OnBootSec=120min
OnUnitActiveSec =
OnUnitActiveSec=120min
```

The changes will be effective after a reload of the daemon with `systemctl daemon-reload` or after a reboot.

```
$ sudo systemctl daemon-reload
$ sudo systemctl list-timers
```

You can get more details about the timer status itself as an administrator:

```
$ sudo systemctl status adsys-gpo-refresh.timer
adsys-gpo-refresh.timer - Refresh ADSys GPO for machine and users
Loaded: loaded (/lib/systemd/system/adsys-gpo-refresh.timer; enabled; vendor preset: enabled)
Active: active (waiting) since Tue 2021-05-18 08:35:48 CEST; 1h 23min ago
Trigger: Tue 2021-05-18 10:05:49 CEST; 6min left
Triggers: adsys-gpo-refresh.service
may 18 08:35:48 adclient04 systemd[1]: Started Refresh ADSys GPO for machine and users.
```

TODO: adsysctl service status to get next scheduled refresh
3.2.2 Socket activation

The ADSys daemon is started on demand by systemd’s socket activation and only runs when it's required. It will gracefully shutdown after idling for a short period of time (by default 120 seconds).

3.2.3 Configuration

ADSys doesn’t ship a configuration file by default. However, such a file can be created to modify the behavior of the daemon and the client.

The configuration file can be system-wide in `/etc/adsys.yaml`. This will thus apply to both daemon and client. You can have per-user configuration by creating `$HOME/adsys.yaml`. It will then affect only the client for this user.

The current directory is also searched for an `adsys.yaml` file.

Finally, each `adsysd` and `adsysctl` commands accept a `--config|-c <configuration_file_path>` flag to set the path to a configuration file at run time. It can be used for testing purpose for instance.

An example of configuration file with all the options can be found in the ADSys repository:

```
# Service and client configuration
verbose: 2
socket: /tmp/adsysd/socket

# Service only configuration
service_timeout: 3600
cache_dir: /tmp/adsysd/cache
run_dir: /tmp/adsysd/run

# Backend selection: sssd (default) or winbind
ad_backend: sssd

# SSSD configuration
sssd:
  config: /etc/sssd.conf
  cache_dir: /var/lib/sss/db

# Winbind configuration
# (if ad_backend is set to winbind)
winbind:
  ad_domain: domain.com
  ad_server: adc.domain.com

# Client only configuration
client_timeout: 60
```
Active Directory GPO support

Configuration common between service and client

- **verbose** Increase the verbosity of the daemon or client. By default, only warnings and error logs are printed. This value is set between 0 and 3. This has the same effect as the -v and -vv flags.

- **socket** Path the Unix socket for communication between clients and daemon. This can be overridden by the --socket option. Defaults to /run/adsysd.sock (monitored by systemd for socket activation).

Service only configuration

- **service_timeout** Time in seconds without any active request before the service exits. This can be overridden by the --timeout option. Defaults to 120 seconds.

- **backend** Backend to use to integrate with Active Directory. It is responsible for providing valid kerberos tickets. Available selection is sssd or winbind. Default is sssd. This can be overridden by the --backend option.

- **sss_cache_dir** The directory that stores Kerberos tickets used by SSSD. By default /var/lib/sss/db/.

- **run_dir** The run directory contains the links to the kerberos tickets for the machine and the active users. This can be overridden by the --run-dir option. Defaults to /run/adsys/.

Backend specific options

**SSSD**

- **config**
  
  Path sssd.conf. This is the source of selected sss domain (first entry in domains:), to find corresponding active directory domain section.

  The option ad_domain in that section is used for the list of domains list of the host. ad_server (optional) is used as the Active directory LDAP server to contact. If it is missing, then the “Active Server” detected by sssd will be used.

  Finally default_domain_suffix is used too, and falls back to the domain name if missing.

  Default lookup path is /etc/sssd/sssd.conf. This can be overridden by the --sssd.config option.

  - **cache_dir**
    
    Path to the sss database to find the HOST kerberos ticket. Default path is /var/lib/sss/db/. This can be overridden by the --sssd.cache-dir option.

**Winbind**

- **ad_domain**
  
  A custom domain can be used to override the C API call that ADSys executes to determine the active domain – which is returned by the wbinfo --own-domain (e.g. example.com)

  - **ad_server**
    
    A custom domain controller can be used to override the C API call that ADSys executes to determine the AD controller FQDN – which is returned by wbinfo --dsgetdcname domain.com (e.g. adc.example.com).
Active Directory GPO support

**Client only configuration:**

- **client_timeout** Maximum time in seconds between 2 server activities before the client returns and aborts the request. This can be overridden by the `--timeout` option. Defaults to 30 seconds.

3.2.4 Debugging with logs (cat command)

It is possible to follow the exchanges between all clients and the daemon with the `cat` command. It forwards all logs and message printing from the daemon alone.

Only privileged users have access to this information. As with any other command, the verbosity can be increased with `-v` flags (it’s independent of the daemon or client current verbosity). More flags increases the verbosity further up to 3. More information is available in the *next chapter* covering `adsysctl` `cat` command.

3.2.5 Authorizations

**ADSys** uses a privilege mechanism based on polkit to manage authorizations. Many commands require elevated privileges to be executed. If the `adsys` client is executed with insufficient privileges to execute a command, the user will be prompted to enter its password. If allowed then the command will be executed and denied otherwise.

This is configurable by the administrator as any service controlled by polkit. For more information man `polkit`.

3.2. The adsys daemon
3.2.6 Additional notes

There are additional configuration options matching the adsysd command line options. Those are used to define things like dconf, apparmor, polkit, sudo directories... Even though they exist mostly for integration tests purposes, they can be tweaked the same way as other configuration options for the service.

3.2.7 Reference guide

Do not hesitate to use the shell completion and the help subcommands to get more information about the subcommands.

3.3 Active Directory Watch Daemon

3.3.1 Monitoring the application

adwatchd is configured to log to the Windows Event Log, and can be monitored using the Event Viewer. By default, the application will only log events when it starts or stops, but the verbose level can be increased via the configuration file to log more information such as files being watched, or the GPT.ini file being updated.

3.3.2 CLI usage

For more advanced usage, the application can be managed from the command line. If the application was installed via the bespoke installer, a helpful shortcut is available in the Start Menu: Start Command Prompt with adwatchd. This will start a Command Prompt window with the adwatchd executable in the PATH.

There are two commands available:

- The run command starts the directory watch loop in foreground mode. This is useful for debugging purposes, as it can be called with the same arguments as the service.
- The service provides a set of subcommands to manage the service.

For detailed descriptions and configuration options of adwatchd, refer to the Command line reference section.

Command line

3.4 adsysctl command line

3.4.1 User commands

adsysctl

AD integration client
Active Directory GPO support

Synopsis

Active Directory integration bridging toolset command line tool.

adsysctl COMMAND [flags]

Options

- `--config` string use a specific configuration file
- `--help` help for adsysctl
- `--socket` string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- `--timeout` int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `--verbose` count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

adsysctl applied

Print last applied GPOs for current or given user/machine

Synopsis

Alias of “policy applied”

adsysctl applied [USER_NAME] [flags]

Options

- `--all` show overridden rules in each GPOs.
- `--details` show applied rules in addition to GPOs.
- `--machine` show applied rules to the machine.
- `--no-color` don't display colorized version.

Options inherited from parent commands

- `--config` string use a specific configuration file
- `--socket` string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- `--timeout` int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `--verbose` count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
**Adsystl completion**

Generate the autocompletion script for the specified shell

**Synopsis**

Generate the autocompletion script for adsystl for the specified shell. See each sub-command’s help for details on how to use the generated script.

**Options**

- `-h`, `--help` help for completion

**Options inherited from parent commands**

- `-c`, `--config` string use a specific configuration file
- `-s`, `--socket` string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsystl.sock")
- `-t`, `--timeout` int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `-v`, `--verbose` count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

**Adsystl completion bash**

Generate the autocompletion script for bash

**Synopsis**

Generate the autocompletion script for the bash shell.

This script depends on the ‘bash-completion’ package. If it is not installed already, you can install it via your OS’s package manager.

To load completions in your current shell session:

```
source <(adsystl completion bash)
```

To load completions for every new session, execute once:
Active Directory GPO support

Linux:

```
adsysctl completion bash > /etc/bash_completion.d/adsysctl
```

macOS:

```
adsysctl completion bash > $(brew --prefix)/etc/bash_completion.d/adsysctl
```

You will need to start a new shell for this setup to take effect.

```
adsysctl completion bash
```

**Options**

```
-h, --help  help for bash
--no-descriptions  disable completion descriptions
```

**Options inherited from parent commands**

```
-c, --config string  use a specific configuration file
-s, --socket string  socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int  time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

**adsysctl completion fish**

Generate the autocompletion script for fish

**Synopsis**

Generate the autocompletion script for the fish shell.

To load completions in your current shell session:

```
adsysctl completion fish | source
```

To load completions for every new session, execute once:

```
adsysctl completion fish > ~/.config/fish/completions/adsysctl.fish
```

You will need to start a new shell for this setup to take effect.

```
adsysctl completion fish [flags]
```

3.4. adsysctl command line
Options

- **-h, --help** help for fish
- **--no-descriptions** disable completion descriptions

Options inherited from parent commands

- **-c, --config string** use a specific configuration file
- **-s, --socket string** socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- **-t, --timeout int** time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- **-v, --verbose count** issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

**adsysctl completion powershell**

Generate the autocompletion script for powershell

**Synopsis**

Generate the autocompletion script for powershell.

To load completions in your current shell session:

```
adsysctl completion powershell | Out-String | Invoke-Expression
```

To load completions for every new session, add the output of the above command to your powershell profile.

```
adsysctl completion powershell [flags]
```

Options

- **-h, --help** help for powershell
- **--no-descriptions** disable completion descriptions

Options inherited from parent commands

- **-c, --config string** use a specific configuration file
- **-s, --socket string** socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- **-t, --timeout int** time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- **-v, --verbose count** issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
adsysctl completion zsh

Generate the autocompletion script for zsh

**Synopsis**

Generate the autocompletion script for the zsh shell.

If shell completion is not already enabled in your environment you will need to enable it. You can execute the following once:

```
echo "autoload -U compinit; compinit" >> ~/.zshrc
```

To load completions in your current shell session:

```
source <(adsysctl completion zsh)
```

To load completions for every new session, execute once:

**Linux:**

```
adsysctl completion zsh > "${fpath[1]}/_adsysctl"
```

**macOS:**

```
adsysctl completion zsh > $(brew --prefix)/share/zsh/site-functions/_adsysctl
```

You will need to start a new shell for this setup to take effect.

```
adsysctl completion zsh [flags]
```

**Options**

```
-h, --help help for zsh
--no-descriptions disable completion descriptions
```

**Options inherited from parent commands**

```
-c, --config string use a specific configuration file
-s, --socket string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

3.4. adsysctl command line
Active Directory GPO support

adsysctl doc

Documentation

```
adsysctl doc [CHAPTER] [flags]
```

**Options**

```
-h, --help    help for doc
```

**Options inherited from parent commands**

```
-c, --config string     use a specific configuration file
-s, --socket string    socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int       time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count     issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adsysctl policy

Policy management

```
adsysctl policy COMMAND [flags]
```

**Options**

```
-h, --help    help for policy
```

**Options inherited from parent commands**

```
-c, --config string     use a specific configuration file
-s, --socket string    socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int       time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count     issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
**adsysctl policy admx**

Dump windows policy definitions

```
adsysctl policy admx lts-only|all [flags]
```

**Options**

```
--distro string distro for which to retrieve policy definition. (default "Ubuntu")
-h, --help help for admx
```

**Options inherited from parent commands**

```
-c, --config string use a specific configuration file
-s, --socket string socket path to use between daemon and client. Can be overridden by systemctl socket activation. (default "/run/adsysd.sock")
-t, --timeout int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

**adsysctl policy applied**

Print last applied GPOs for current or given user/machine

```
adsysctl policy applied [USER_NAME] [flags]
```

**Options**

```
-a, --all show overridden rules in each GPOs.
--details show applied rules in addition to GPOs.
-h, --help help for applied
-m, --machine show applied rules to the machine.
--no-color don't display colorized version.
```

**Options inherited from parent commands**

```
-c, --config string use a specific configuration file
-s, --socket string socket path to use between daemon and client. Can be overridden by systemctl socket activation. (default "/run/adsysd.sock")
-t, --timeout int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
Active Directory GPO support

**adsysctl policy purge**

Purges policies for the current user or a specified one

```
adsysctl policy purge [USER_NAME] [flags]
```

**Options**

- **-a, --all** all purges the policy of the computer and all the logged in users. -m, or USER_NAME cannot be used with this option.
- **-h, --help** help for purge
- **-m, --machine** machine purges the policy of the computer.

**Options inherited from parent commands**

- **-c, --config string** use a specific configuration file
- **-s, --socket string** socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- **-t, --timeout int** time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- **-v, --verbose count** issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

**adsysctl policy update**

Updates/Create a policy for current user or given user with its kerberos ticket

```
adsysctl policy update [USER_NAME KERBEROS_TICKET_PATH] [flags]
```

**Options**

- **-a, --all** all updates the policy of the computer and all the logged in users. -m, or USER_NAME/TICKET cannot be used with this option.
- **-h, --help** help for update
- **-m, --machine** machine updates the policy of the computer.

**Options inherited from parent commands**

- **-c, --config string** use a specific configuration file
- **-s, --socket string** socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- **-t, --timeout int** time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- **-v, --verbose count** issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
adsysctl service

Service management

```
adsysctl service COMMAND [flags]
```

Options

```
-h, --help  help for service
```

Options inherited from parent commands

```
-c, --config string  use a specific configuration file
-s, --socket string  socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int    time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adsysctl service cat

Print service logs

```
adsysctl service cat [flags]
```

Options

```
-h, --help  help for cat
```

Options inherited from parent commands

```
-c, --config string  use a specific configuration file
-s, --socket string  socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int    time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
Active Directory GPO support

**adsysctl service status**

Print service status

```
adsysctl service status [flags]
```

**Options**

- `-h, --help` help for status

**Options inherited from parent commands**

- `-c, --config string` use a specific configuration file
- `-s, --socket string` socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- `-t, --timeout int` time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `-v, --verbose count` issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

**adsysctl service stop**

Requests to stop the service once all connections are done

```
adsysctl service stop [flags]
```

**Options**

- `-f, --force` force will shut it down immediately and drop existing connections.
- `-h, --help` help for stop

**Options inherited from parent commands**

- `-c, --config string` use a specific configuration file
- `-s, --socket string` socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- `-t, --timeout int` time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `-v, --verbose count` issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
adsysctl update

Updates/Create a policy for current user or given user with its kerberos ticket

Synopsis

Alias of “policy update”

```
adsysctl update [USER_NAME KERBEROS_TICKET_PATH] [flags]
```

Options

```
-a, --all all updates the policy of the computer and all the logged in users. -m,
-or USER_NAME/TICKET cannot be used with this option.
-h, --help help for update
-m, --machine machine updates the policy of the computer.
```

Options inherited from parent commands

```
-c, --config string use a specific configuration file
-s, --socket string socket path to use between daemon and client. Can be overridden
-by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int time in seconds before cancelling the client request when the
-server gives no result. 0 for no timeout. (default 30)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adsysctl version

Returns version of client and service

```
adsysctl version [flags]
```

Options

```
-h, --help help for version
```

Options inherited from parent commands

```
-c, --config string use a specific configuration file
-s, --socket string socket path to use between daemon and client. Can be overridden
-by systemd socket activation. (default "/run/adsysd.sock")
-t, --timeout int time in seconds before cancelling the client request when the
-server gives no result. 0 for no timeout. (default 30)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

3.4. adsysctl command line
### 3.4.2 Hidden commands

Those commands are hidden from help and should primarily be used by the system or for debugging.

**adsysctl policy debug**

Debug various policy infos

```bash
adsysctl policy debug [flags]
```

**Options**

- `-h, --help` help for debug

**Options inherited from parent commands**

- `-c, --config string` use a specific configuration file
- `-s, --socket string` socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- `-t, --timeout int` time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `-v, --verbose count` issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

**adsysctl policy debug cert-autoenroll-script**

Write certificate autoenrollment python embedded script in current directory

```bash
adsysctl policy debug cert-autoenroll-script [flags]
```

**Options**

- `-h, --help` help for cert-autoenroll-script

**Options inherited from parent commands**

- `-c, --config string` use a specific configuration file
- `-s, --socket string` socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- `-t, --timeout int` time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- `-v, --verbose count` issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
Active Directory GPO support

adsysctl policy debug gpolist-script

Write GPO list python embedded script in current directory

adsysctl policy debug gpolist-script [flags]

Options

- h, --help help for gpolist-script

Options inherited from parent commands

- c, --config string use a specific configuration file
- s, --socket string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- t, --timeout int time in seconds before cancelling the client request when the server gives no result. 0 for no timeout. (default 30)
- v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

3.5 adsysd command line

3.5.1 User commands

adsysd

AD integration daemon

Synopsis

Active Directory integration bridging toolset daemon.

adsysd COMMAND [flags]

Options

- ad-backend string Active Directory authentication backend (default "sssd")
- cache-dir string directory where AD sys caches GPOs downloads and policies. (default "/var/cache/adsys")
- c, --config string use a specific configuration file
- h, --help help for adsysd
- run-dir string directory where AD sys stores transient information erased on reboot. (default "/run/adsys")
- s, --socket string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
- sssd.cache-dir string SSSd cache directory (default "/var/lib/sss/db")

(continues on next page)
### Active Directory GPO support

(continued from previous page)

```
--sssd.config string  SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int     time in seconds without activity before the service exists. 0 for no timeout. (default 120)
-v, --verbose count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```

### adsysd completion

Generate the autocompletion script for the specified shell

#### Synopsis

Generate the autocompletion script for adsysd for the specified shell. See each sub-command’s help for details on how to use the generated script.

#### Options

```
-h, --help      help for completion
```

#### Options inherited from parent commands

```
--ad-backend string  Active Directory authentication backend (default "sssd")
--cache-dir string   directory where ADSys caches GPOs downloads and policies. (default "/var/cache/adsys")
-c, --config string  use a specific configuration file
--run-dir string     directory where ADSys stores transient information erased on reboot. (default "/run/adsys")
-s, --socket string  socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string  SSSd cache directory (default "/var/lib/sss/db")
--sssd.config string  SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int    time in seconds without activity before the service exists. 0 for no timeout. (default 120)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```

### adsysd completion bash

Generate the autocompletion script for bash
Synopsis

Generate the autocompletion script for the bash shell.

This script depends on the ‘bash-completion’ package. If it is not installed already, you can install it via your OS’s package manager.

To load completions in your current shell session:

```
source <(adsysd completion bash)
```

To load completions for every new session, execute once:

**Linux:**

```
adsysd completion bash > /etc/bash_completion.d/adsysd
```

**macOS:**

```
adsysd completion bash > $(brew --prefix)/etc/bash_completion.d/adsysd
```

You will need to start a new shell for this setup to take effect.

```
adsysd completion bash
```

Options

```
-h, --help help for bash
--no-descriptions disable completion descriptions
```

Options inherited from parent commands

```
--ad-backend string Active Directory authentication backend (default "sssd")
--cache-dir string directory where ADSys caches GPOs downloads and policies.
\(\text{default} \"/var/cache/adsys\")
-c, --config string use a specific configuration file
--run-dir string directory where ADSys stores transient information erased on reboot. \(\text{default} \"/run/adsys\")
-s, --socket string socket path to use between daemon and client. Can be overridden by systemd socket activation. \(\text{default} \"/run/adsysd.sock\")
--sssd.cache-dir string SSSd cache directory \(\text{default} \"/var/lib/sss/db\")
--sssd.config string SSSd config file path \(\text{default} \"/etc/sssd/sssd.conf\")
-t, --timeout int time in seconds without activity before the service exists. \(0 \text{ for no timeout.} \text{default} 120\)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```
Active Directory GPO support

adsysd completion fish

Generate the autocompletion script for fish

Synopsis

Generate the autocompletion script for the fish shell.

To load completions in your current shell session:

```
adsysd completion fish | source
```

To load completions for every new session, execute once:

```
adsysd completion fish > ~/.config/fish/completions/adsysd.fish
```

You will need to start a new shell for this setup to take effect.

```
adsysd completion fish [flags]
```

Options

```
-h, --help          help for fish
--no-descriptions  disable completion descriptions
```

Options inherited from parent commands

```
--ad-backend string  Active Directory authentication backend (default "sssd")
--cache-dir string   directory where ADSys caches GPOs downloads and policies. (default "/var/cache/adsys")
-c, --config string  use a specific configuration file
--run-dir string     directory where ADSys stores transient information erased on reboot. (default "/run/adsys")
-s, --socket string  socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string  SSSd cache directory (default "/var/lib/sssd/db")
--sssd.config string  SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int    time in seconds without activity before the service exists. 0 for no timeout. (default 120)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
**adsysd completion powershell**

Generate the autocompletion script for powershell

**Synopsis**

Generate the autocompletion script for powershell.

To load completions in your current shell session:

```
adsysd completion powershell | Out-String | Invoke-Expression
```

To load completions for every new session, add the output of the above command to your powershell profile.

```
adsysd completion powershell [flags]
```

**Options**

```
-h, --help help for powershell
--no-descriptions disable completion descriptions
```

**Options inherited from parent commands**

```
--ad-backend string Active Directory authentication backend (default "sssd")
--cache-dir string directory where ADSys caches GPOs downloads and policies.
  (default "/var/cache/adsys")
-c, --config string use a specific configuration file
--run-dir string directory where ADSys stores transient information.
  erased on reboot. (default "/run/adsys")
-s, --socket string socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string SSSd cache directory (default "/var/lib/sss/db")
--sssd.config string SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int time in seconds without activity before the service exists. 0 for no timeout. (default 120)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```

**adsysd completion zsh**

Generate the autocompletion script for zsh
Active Directory GPO support

Synopsis

Generate the autocompletion script for the zsh shell.

If shell completion is not already enabled in your environment you will need to enable it. You can execute the following once:

```
echo "autoload -U compinit; compinit" >> ~/.zshrc
```

To load completions in your current shell session:

```
source <(adsysd completion zsh)
```

To load completions for every new session, execute once:

### Linux:

```
adsysd completion zsh > "${fpath[1]}/_adsysd"
```

### macOS:

```
adsysd completion zsh > $(brew --prefix)/share/zsh/site-functions/_adsysd
```

You will need to start a new shell for this setup to take effect.

```
adsysd completion zsh [flags]
```

Options

```
-h, --help help for zsh
--no-descriptions disable completion descriptions
```

Options inherited from parent commands

```
--ad-backend string Active Directory authentication backend (default "sssd")
--cache-dir string directory where ADSys caches GPOs downloads and policies.
( default "/var/cache/adsys")
-c, --config string use a specific configuration file
--run-dir string directory where ADSys stores transient information
  erased on reboot. (default "/run/adsys")
-s, --socket string socket path to use between daemon and client. Can be
  overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string SSSd cache directory (default "/var/lib/sss/db")
--sssd.config string SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int time in seconds without activity before the service
  exists. 0 for no timeout. (default 120)
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```
adsysd version

Returns version of service and exits

```
adsysd version [flags]
```

Options

```
-h, --help  help for version
```

Options inherited from parent commands

```
--ad-backend string  Active Directory authentication backend (default "sssd")
--cache-dir string  directory where ADSys caches GPOs downloads and policies.
 (default "/var/cache/adsys")
-c, --config string  use a specific configuration file
--run-dir string  directory where ADSys stores transient information erased on reboot. (default "/run/adsys")
-s, --socket string  socket path to use between daemon and client. Can be overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string  SSSd cache directory (default "/var/lib/sss/db")
--sssd.config string  SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int  time in seconds without activity before the service exists. 0 for no timeout. (default 120)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```

3.5.2 Hidden commands

Those commands are hidden from help and should primarily be used by the system or for debugging.

adsysd mount

Mount the locations listed in the specified file for the current user

```
adsysd mount MOUNTS_FILE [flags]
```

Options

```
-h, --help  help for mount
```
Active Directory GPO support

Options inherited from parent commands

```
--ad-backend string   Active Directory authentication backend (default "sssd")
--cache-dir string   directory where ADSys caches GPOs downloads and policies.
   (default "/var/cache/adsys")
-c, --config string  use a specific configuration file
--run-dir string     directory where ADSys stores transient information
                       erased on reboot. (default "/run/adsys")
-s, --socket string  socket path to use between daemon and client. Can be
                       overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string SSSd cache directory (default "/var/lib/sss/db")
--sssd.config string SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int    time in seconds without activity before the service
                       exists. 0 for no timeout. (default 120)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```

adsysd runscripts

Runs scripts in the given subdirectory

```
adsysd runscripts ORDER_FILE [flags]
```

Options

```
   --allow-order-missing allow ORDER_FILE to be missing once the scripts are ready.
-h, --help help for runscripts
```

Options inherited from parent commands

```
--ad-backend string   Active Directory authentication backend (default "sssd")
--cache-dir string   directory where ADSys caches GPOs downloads and policies.
   (default "/var/cache/adsys")
-c, --config string  use a specific configuration file
--run-dir string     directory where ADSys stores transient information
                       erased on reboot. (default "/run/adsys")
-s, --socket string  socket path to use between daemon and client. Can be
                       overridden by systemd socket activation. (default "/run/adsysd.sock")
--sssd.cache-dir string SSSd cache directory (default "/var/lib/sss/db")
--sssd.config string SSSd config file path (default "/etc/sssd/sssd.conf")
-t, --timeout int    time in seconds without activity before the service
                       exists. 0 for no timeout. (default 120)
-v, --verbose count  issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv)
```

60 Chapter 3. Technical Reference
3.6 adwatchd command line

3.6.1 User commands

adwatchd

AD watch daemon

Synopsis

Watch directories for changes and bump the relevant GPT.ini versions.

```
adwatchd [COMMAND] [flags]
```

Options

```
-c, --config string use a specific configuration file
-h, --help help for adwatchd
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adwatchd completion

Generate the autocompletion script for the specified shell

Synopsis

Generate the autocompletion script for adwatchd for the specified shell. See each sub-command’s help for details on how to use the generated script.

Options

```
-h, --help help for completion
```

Options inherited from parent commands

```
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adwatchd completion bash

Generate the autocompletion script for bash

**Synopsis**

Generate the autocompletion script for the bash shell.

This script depends on the ‘bash-completion’ package. If it is not installed already, you can install it via your OS’s package manager.

To load completions in your current shell session:

```bash
source <(adwatchd completion bash)
```

To load completions for every new session, execute once:

**Linux:**

```bash
adwatchd completion bash > /etc/bash_completion.d/adwatchd
```

**macOS:**

```bash
adwatchd completion bash > $(brew --prefix)/etc/bash_completion.d/adwatchd
```

You will need to start a new shell for this setup to take effect.

adwatchd completion bash

**Options**

```
-h, --help        help for bash
--no-descriptions disable completion descriptions
```

**Options inherited from parent commands**

```
-v, --verbose     count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
**adwatchd completion fish**

Generate the autocompletion script for fish

**Synopsis**

Generate the autocompletion script for the fish shell.

To load completions in your current shell session:

```
adwatchd completion fish | source
```

To load completions for every new session, execute once:

```
adwatchd completion fish > ~/.config/fish/completions/adwatchd.fish
```

You will need to start a new shell for this setup to take effect.

```
adwatchd completion fish [flags]
```

**Options**

```
-h, --help help for fish
--no-descriptions disable completion descriptions
```

**Options inherited from parent commands**

```
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

**adwatchd completion powershell**

Generate the autocompletion script for powershell

**Synopsis**

Generate the autocompletion script for powershell.

To load completions in your current shell session:

```
adwatchd completion powershell | Out-String | Invoke-Expression
```

To load completions for every new session, add the output of the above command to your powershell profile.

```
adwatchd completion powershell [flags]
```
**Active Directory GPO support**

Options

- **-h, --help** help for powershell
- **--no-descriptions** disable completion descriptions

Options inherited from parent commands

- **-v, --verbose** count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

adwatchd completion zsh

Generate the autocompletion script for zsh

**Synopsis**

Generate the autocompletion script for the zsh shell.

If shell completion is not already enabled in your environment you will need to enable it. You can execute the following once:

```
echo "autoload -U compinit; compinit" >> ~/.zshrc
```

To load completions in your current shell session:

```
source <(adwatchd completion zsh)
```

To load completions for every new session, execute once:

**Linux:**

```
adwatchd completion zsh > "${fpath[1]}/_adwatchd"
```

**macOS:**

```
adwatchd completion zsh > $(brew --prefix)/share/zsh/site-functions/_adwatchd
```

You will need to start a new shell for this setup to take effect.

```
adwatchd completion zsh [flags]
```
Active Directory GPO support

Options

-h, --help
help for zsh
-no-descriptions
disable completion descriptions

Options inherited from parent commands

-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

adwatchd run

Starts the directory watch loop

Synopsis

Can run as a service through the service manager or interactively as a standalone application.

The program will monitor the configured directories for changes and bump the appropriate GPT.ini versions anytime a change is detected. If a GPT.ini file does not exist for a directory, a warning will be issued and the file will be created. If the GPT.ini file is incompatible or malformed, the program will report an error.

adwatchd run [flags]

Options

-c, --config string use a specific configuration file
-d, --dirs directory a directory to check for changes (can be specified multiple times)
-f, --force force the program to run even if another instance is already running
-h, --help help for run

Options inherited from parent commands

-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

adwatchd service

Manages the adwatchd service

3.6. adwatchd command line
Active Directory GPO support

Synopsis

The service command allows the user to interact with the adwatchd service. It can manage and query the service status, and also install and uninstall the service.

```
adwatchd service COMMAND [flags]
```

Options

```
-h, --help help for service
```

Options inherited from parent commands

```
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adwatchd service install

Installs the service

Synopsis

Installs the adwatchd service.

The service will be installed as a Windows service.

```
adwatchd service install [flags]
```

Options

```
-c, --config string use a specific configuration file
-h, --help help for install
```

Options inherited from parent commands

```
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
**adwatchd service restart**

Restarts the service

**Synopsis**

Restarts the adwatchd service.

```
adwatchd service restart [flags]
```

**Options**

```
-h, --help   help for restart
```

**Options inherited from parent commands**

```
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

**adwatchd service start**

Starts the service

**Synopsis**

Starts the adwatchd service.

```
adwatchd service start [flags]
```

**Options**

```
-h, --help   help for start
```

**Options inherited from parent commands**

```
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```
Active Directory GPO support

adwatchd service status
Returns service status

Synopsis
Returns the status of the adwatchd service.
adwatchd service status [flags]

Options
-h, --help help for status

Options inherited from parent commands
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output

adwatchd service stop
Stops the service

Synopsis
Stops the adwatchd service.
adwatchd service stop [flags]

Options
-h, --help help for stop

Options inherited from parent commands
-v, --verbose count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
adwatchd service uninstall

Uninstalls the service

Synopsis

Uninstalls the adwatchd service.

```
adwatchd service uninstall [flags]
```

Options

```
-h, --help        help for uninstall
```

Options inherited from parent commands

```
-v, --verbose     count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

adwatchd version

Returns version of service and exits

```
adwatchd version [flags]
```

Options

```
-h, --help        help for version
```

Options inherited from parent commands

```
-v, --verbose     count issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

3.6.2 Hidden commands

Those commands are hidden from help and should primarily be used by the system or for debugging.
Supported policies

3.7 Policies reference list

Here is the list of each supported policies by ADSys. The categories and elements are under the same hierarchy you will find on your Active Directory controller.

3.7.1 Computer Policies

Ubuntu

Client management

Computer Scripts

Shutdown scripts

Define scripts that are executed on machine power off. Those scripts are ordered, one by line, and relative to SYSVOL/ubuntu/scripts/ directory. Scripts from this GPO will be appended to the list of scripts referenced higher in the GPO hierarchy.

- Type: scripts
- Key: /shutdown

Note: -
- Enabled: The scripts in the text entry are executed at shutdown time.
- Disabled: The scripts will be skipped. The set of scripts are per boot, and refreshed only on new boot of the machine.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Computer Scripts -&gt; Shutdown scripts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\scripts\shutdown</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
**Startup scripts**

Define scripts that are executed on machine boot, once the GPO is downloaded. Those scripts are ordered, one by line, and relative to SYSVOL/ubuntu/scripts/ directory. Scripts from this GPO will be appended to the list of scripts referenced higher in the GPO hierarchy.

- **Type:** scripts
- **Key:** /startup

**Note:**
- **Enabled:** The scripts in the text entry are executed at startup time.
- **Disabled:** The scripts will be skipped. The set of scripts are per boot, and refreshed only on new boot of the machine.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Computer Scripts -&gt; Startup scripts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\scripts\startup</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**Power Management**

**Enable the ALS sensor**

If the ambient light sensor functionality is enabled.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/ambient-enabled
- **Default:** true

**Note:** default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Enable the ALS sensor</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\ambient-enabled</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Active Directory GPO support

The brightness of the screen when idle

This is the laptop panel screen brightness used when the session is idle.

- Type: dconf
- Key: /org/gnome/settings-daemon/plugins/power/idle-brightness
- Default: 30

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; The brightness of the screen when idle</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\idle-brightness</td>
</tr>
<tr>
<td>Element type</td>
<td>decimal</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Dim the screen after a period of inactivity

If the screen should be dimmed to save power when the computer is idle.

- Type: dconf
- Key: /org/gnome/settings-daemon/plugins/power/idle-dim
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Dim the screen after a period of inactivity</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\idle-dim</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Laptop lid close action when on AC

The action to take when the laptop lid is closed and the laptop is on AC power.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/lid-close-ac-action
- **Default:** ‘suspend’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Valid values**

- blank
- suspend
- shutdown
- hibernate
- interactive
- nothing
- logout

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Laptop lid close action when on AC</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\lid-close-ac-action</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Laptop lid close action on battery

The action to take when the laptop lid is closed and the laptop is on battery power.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/lid-close-battery-action
- **Default:** ‘suspend’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Valid values**

- blank
- suspend
- shutdown
Active Directory GPO support

- hibernate
- interactive
- nothing
- logout

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Laptop lid close action on battery</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\lid-close-battery-action</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Laptop lid, when closed, will suspend even if there is an external monitor plugged in

With no external monitors plugged in, closing a laptop’s lid will suspend the machine (as set by the lid-close-battery-action and lid-close-ac-action keys). By default, however, closing the lid when an external monitor is present will not suspend the machine, so that one can keep working on that monitor (e.g. for docking stations or media viewers). Set this key to False to keep the default behavior, or to True to suspend the laptop whenever the lid is closed and regardless of external monitors.

- Type: dconf
- Key: /org/gnome/settings-daemon/plugins/power/lid-close-suspend-with-external-monitor
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Laptop lid, when closed, will suspend even if there is an external monitor plugged in</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\lid-close-suspend-with-external-monitor</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
**Power button action**

The action to take when the system power button is pressed. Virtual machines only honor the ‘nothing’ action, and will shutdown otherwise. Tablets always suspend, ignoring all the other action options.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/power-button-action
- **Default:** ‘interactive’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Valid values**

- nothing
- suspend
- hibernate
- interactive

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Power button action</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\power-button-action</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**Enable power-saver profile when battery is low**

Automatically enable the “power-saver” profile using power-profiles-daemon if the battery is low.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/power-saver-profile-on-low-battery
- **Default:** true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Enable power-saver profile when battery is low</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\power-saver-profile-on-low-battery</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Active Directory GPO support

Sleep timeout computer when on AC

The amount of time in seconds the computer on AC power needs to be inactive before it goes to sleep. A value of 0 means never.

- Type: dconf
- Key: /org/gnome/settings-daemon/plugins/power/sleep-inactive-ac-timeout
- Default: 0

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Sleep timeout computer when on AC</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\sleep-inactive-ac-timeout</td>
</tr>
<tr>
<td>Element type</td>
<td>decimal</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Whether to hibernate, suspend or do nothing when inactive

The type of sleeping that should be performed when the computer is inactive.

- Type: dconf
- Key: /org/gnome/settings-daemon/plugins/power/sleep-inactive-ac-type
- Default: ‘suspend’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Valid values

- blank
- suspend
- shutdown
- hibernate
- interactive
- nothing
- logout

Metadata
### Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Whether to hibernate, suspend or do nothing when inactive</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\sleep-inactive-ac-type</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

### Sleep timeout computer when on battery

The amount of time in seconds the computer on battery power needs to be inactive before it goes to sleep. A value of 0 means never.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/sleep-inactive-battery-timeout
- **Default for 20.04:** 1200
- **Default for 22.04:** 1200
- **Default for 23.04:** 900
- **Default for 23.10:** 900

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Sleep timeout computer when on battery</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\sleep-inactive-battery-timeout</td>
</tr>
<tr>
<td>Element type</td>
<td>decimal</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

### Whether to hibernate, suspend or do nothing when inactive

The type of sleeping that should be performed when the computer is inactive.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/power/sleep-inactive-battery-type
- **Default:** ‘suspend’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Valid values
Active Directory GPO support

- blank
- suspend
- shutdown
- hibernate
- interactive
- nothing
- logout

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Power Management -&gt; Whether to hibernate, suspend or do nothing when inactive</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\power\sleep-inactive-battery-type</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Privilege Authorization

Allow local administrators

This allows or prevents client machine to have local users gaining administrators privilege on the machine.

- Type: privilege
- Key: /allow-local-admins

Note: -

- Enabled: This leaves the default rules for the “sudo” and “admin” rule intact.
- Disabled: This denies root privileges to the predefined administrator groups (sudo and admin).

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Privilege Authorization -&gt; Allow local administrators</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\privilege\allow-local-admins</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
**Client administrators**

Define users and groups from AD allowed to administer client machines. It must be of the form user@domain or %group@domain. One per line.

- **Type:** privilege
- **Key:** /client-admins

**Note:**
- **Enabled:** This allows defining Active Directory groups and users with administrative privileges in the box entry.
- **Disabled:** This disallows any Active Directory group or user to become an administrator of the client even if it is defined in a parent GPO of the hierarchy tree.

An Ubuntu Pro subscription on the client is required to apply this policy.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; Privilege Authorization -&gt; Client administrators</td>
</tr>
<tr>
<td>Registry</td>
<td>Software\Policies\Ubuntu\privilege\client-admins</td>
</tr>
<tr>
<td>Key</td>
<td></td>
</tr>
<tr>
<td>Element</td>
<td>multiText</td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>Class</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**System Drive Mapping**

**System mounts**

Define network shares that will be mounted for the system. If more shares are defined higher in the GPO hierarchy, the entries listed here will be appended to the list and duplicates will be removed.

Values should be in the format: :/// e.g. nfs://example_nfs.com/nfs_shared_dir smb://example_smb.com/smb_shared_dir ftp://ftp_share_server.com

This pattern must be followed, otherwise the policy will not be applied.

By default, the mounts will be done in anonymous mode. In case of authentication needed, a krb5 tag can be added to the value, e.g. [krb5]://

If the tag is added, the mount will require Kerberos authentication in order to occur.

The supported protocols / file systems are the same as the ones supported by the mount command. They are listed on the mount man page on https://man7.org/linux/man-pages/man8/mount.8.html It’s up to the user to ensure that the requested protocols are valid and supported and that the shared directories have the correct configuration for the requested connection.

- **Type:** mount
- **Key:** /system-mounts

**Note:**
- **Enabled:** The value(s) referenced in the entry are applied on the client machine.
Active Directory GPO support

- Disabled: The value(s) are removed from the target machine.
- Not configured: Value(s) declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System Drive Mapping -&gt; System mounts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\mount\system-mounts</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

System proxy configuration

Auto-configuration URL

Declare system-wide proxy auto-configuration URL.

Auto-configuration URLs are always prioritized over manual proxy settings, meaning that if all proxy options are set, the GPO client will enable automatic proxy configuration for supported backends. An empty value will remove previously set settings of the same type.

- Type: /proxy/auto

Note:

- Enabled: The setting in the text entry is applied on the client machine.
- Disabled: The setting is removed from the target machine.
- Not configured: A setting declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System proxy configuration -&gt; Auto-configuration URL</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\proxy\proxy\auto</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Active Directory GPO support

FTP Proxy

Declare system-wide HTTPS proxy setting. The value must be in the form of:
protocol://username:password@host:port

It is not mandatory to escape special characters in the username or password. The GPO client will escape any unescaped special character before applying the proxy settings, and will take care not to double-escape already escaped characters. An empty value will remove previously set settings of the same type.

- Type: proxy
- Key: /proxy/ftp

Note:

- Enabled: The setting in the text entry is applied on the client machine.
- Disabled: The setting is removed from the target machine.
- Not configured: A setting declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System proxy configuration -&gt; FTP Proxy</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\proxy\proxy\ftp</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

HTTP Proxy

Declare system-wide HTTP proxy setting. The value must be in the form of:
protocol://username:password@host:port

It is not mandatory to escape special characters in the username or password. The GPO client will escape any unescaped special character before applying the proxy settings, and will take care not to double-escape already escaped characters. An empty value will remove previously set settings of the same type.

- Type: proxy
- Key: /proxy/http

Note:

- Enabled: The setting in the text entry is applied on the client machine.
- Disabled: The setting is removed from the target machine.
- Not configured: A setting declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata
### Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System proxy configuration -&gt; HTTP Proxy</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\proxy\proxy\http</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

#### HTTPS Proxy

Declare system-wide HTTPS proxy setting. The value must be in the form of:

```
protocol://username:password@host:port
```

It is not mandatory to escape special characters in the username or password. The GPO client will escape any unescaped special character before applying the proxy settings, and will take care not to double-escape already escaped characters. An empty value will remove previously set settings of the same type.

- Type: proxy
- Key: /proxy/https

**Note:**
- Enabled: The setting in the text entry is applied on the client machine.
- Disabled: The setting is removed from the target machine.
- Not configured: A setting declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

#### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System proxy configuration -&gt; HTTPS Proxy</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\proxy\proxy\https</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

#### Ignored hosts

An array of hosts allowed to bypass the proxy settings. The host exclusion setting must be in the form of:

```
localhost,127.0.0.1::1
```

Hosts can be individually wrapped in single (’) or double quotes (“”), or separated by spaces. An empty value will remove previously set settings of the same type.

- Type: proxy
- Key: /proxy/no-proxy

**Note:**
- Enabled: The setting in the text entry is applied on the client machine.
Active Directory GPO support

- Disabled: The setting is removed from the target machine.
- Not configured: A setting declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System proxy configuration -&gt; Ignored hosts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\proxy\proxy\no-proxy</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**SOCKS Proxy**

Declare system-wide HTTPS proxy setting. The value must be in the form of:

`protocol://username:password@host:port`

It is not mandatory to escape special characters in the username or password. The GPO client will escape any unescaped special character before applying the proxy settings, and will take care not to double-escape already escaped characters. An empty value will remove previously set settings of the same type.

- Type: proxy
- Key: /proxy/socks

**Note:**

- Enabled: The setting in the text entry is applied on the client machine.
- Disabled: The setting is removed from the target machine.
- Not configured: A setting declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System proxy configuration -&gt; SOCKS Proxy</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\proxy\proxy\socks</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Active Directory GPO support

System-wide application confinement

AppArmor

Define AppArmor profiles to be parsed and loaded on client machines. These profiles are ordered, one by line, and relative to the SYSVOL/ubuntu/apparmor/ directory. On the client machine, computer profiles are stored in /etc/apparmor.d/adsys/machine, thus the administrator can reference abstractions and tunables shipped with the client distribution of AppArmor. Files can be included in each other either using a path relative to the current directory of the profile (include “path/to/profile”), or relying on the include path of AppArmor (include <adsys/machine/path/to/profile>).

Profiles from this GPO will be appended to the list of profiles referenced higher in the GPO hierarchy.

- Type: apparmor
- Key: /apparmor-machine

Note: -
- Enabled: The profiles in the text entry are applied on the client machine.
- Disabled: The profiles are removed from the target machine, and any related rules are unloaded.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Client management -&gt; System-wide application confinement -&gt; AppArmor</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\apparmor\apparmor-machine</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Login Screen

Authentication

Number of allowed authentication failures

The number of times a user is allowed to attempt authentication, before giving up and going back to user selection.

- Type: dconf
- Key: /org/gnome/login-screen/allowed-failures
- Default: 3

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata
### Whether or not to allow fingerprint readers for login

The login screen can optionally allow users who have enrolled their fingerprints to log in using those prints.

- Type: dconf
- Key: /org/gnome/login-screen/enable-fingerprint-authentication
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Authentication -&gt; Whether or not to allow fingerprint readers for login</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\enable-fingerprint-authentication</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

### Whether or not to allow passwords for login

The login screen can be configured to disallow password authentication, forcing the user to use smartcard or fingerprint authentication.

- Type: dconf
- Key: /org/gnome/login-screen/enable-password-authentication
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Authentication -&gt; Whether or not to allow passwords for login</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\enable-password-authentication</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
### Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Location</strong></td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Authentication -&gt; Whether or not to allow passwords for login</td>
</tr>
<tr>
<td><strong>Registry Key</strong></td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\enable-password-authentication</td>
</tr>
<tr>
<td><strong>Element type</strong></td>
<td>boolean</td>
</tr>
<tr>
<td><strong>Class:</strong></td>
<td>Machine</td>
</tr>
</tbody>
</table>

**Whether or not to allow smartcard readers for login**

The login screen can optionally allow users who have smartcards to log in using those smartcards.
- Type: dconf
- Key: /org/gnome/login-screen/enable-smartcard-authentication
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Location</strong></td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Authentication -&gt; Whether or not to allow smartcard readers for login</td>
</tr>
<tr>
<td><strong>Registry Key</strong></td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\enable-smartcard-authentication</td>
</tr>
<tr>
<td><strong>Element type</strong></td>
<td>boolean</td>
</tr>
<tr>
<td><strong>Class:</strong></td>
<td>Machine</td>
</tr>
</tbody>
</table>

**Interface**

**The background-color property sets the background color.**

The background-color property sets the background color to use when the background picture URI is missing or when it doesn’t cover the whole background. It overrides the value defined in the default style sheet.
- Type: dconf
- Key: /com/ubuntu/login-screen/background-color
- Default: ‘’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**
**Active Directory GPO support**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; The background-color property sets the background color.</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\com\ubuntu\login-screen\background-color</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**Sets the background image for the login screen.**

URI to use for the background image. Note that the backend only supports local (file://) URIs. It overrides the value defined in the default style sheet.

- Type: dconf
- Key: /com/ubuntu/login-screen/background-picture-uri
- Default: ‘’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; Sets the background image for the login screen.</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\com\ubuntu\login-screen\background-picture-uri</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**The background-repeat property sets if/how the background image will be repeated.**

The background-repeat property sets if/how a background image will be repeated. By default, a background-image is repeated both vertically and horizontally. It overrides the value defined in the default style sheet.

- Type: dconf
- Key: /com/ubuntu/login-screen/background-repeat
- Default: ‘default’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Valid values**

- default
- repeat
Active Directory GPO support

- repeat-x
- repeat-y
- no-repeat
- space
- round

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; how the background image will be repeated.</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\com\ubuntu\login-screen\background-repeat</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

The background-size property specifies the size of the background image.

The background-size property specifies the size of the background images. There are three keywords you can use with this property: auto: The background image is displayed in its original size; cover: Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges; contain: Resize the background image to make sure the image is fully visible. It overrides the value defined in the default style sheet.

- Type: dconf
- Key: /com/ubuntu/login-screen/background-size
- Default: ‘default’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Valid values

- default
- auto
- cover
- contain

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; The background-size property specifies the size of the background image.</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\com\ubuntu\login-screen\background-size</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Enable showing the banner message

Set to true to show the banner message text.

- Type: dconf
- Key: /org/gnome/login-screen/banner-message-enable
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; Enable showing the banner message</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\banner-message-enable</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Banner message text

Text banner message to show in the login window.

- Type: dconf
- Key: /org/gnome/login-screen/banner-message-text
- Default: ‘’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; Banner message text</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\banner-message-text</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Whether the clock displays in 24h or 12h format

Whether the clock displays in 24h or 12h format

- Type: dconf
- Key: /org/gnome/desktop/interface/clock-format
- Default: ‘24h’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.
Active Directory GPO support

Valid values

- 24h
- 12h

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; Whether the clock displays in 24h or 12h format</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\desktop\interface\clock-format</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Show date in clock

If true, display date in the clock, in addition to time.

- Type: dconf
  - Key: /org/gnome/desktop/interface/clock-show-date
  - Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; Show date in clock</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\desktop\interface\clock-show-date</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Show weekday in clock

If true, display weekday in the clock, in addition to time.

- Type: dconf
  - Key: /org/gnome/desktop/interface/clock-show-weekday
  - Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata
### Path to small image at top of user list

The login screen can optionally show a small image to provide site administrators and distributions a way to display branding.

- **Type**: dconf
- **Key**: `/org/gnome/login-screen/logo`
- **Default**: ‘/usr/share/plymouth/ubuntu-logo.png’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Interface -&gt; Path to small image at top of user list</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\logo</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

### Disable showing the restart buttons

Set to true to disable showing the restart buttons in the login window.

- **Type**: dconf
- **Key**: `/org/gnome/login-screen/disable-restart-buttons`
- **Default**: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Disable showing the restart buttons</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\disable-restart-buttons</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>
Active Directory GPO support

Avoid showing user list

The login screen normally shows a list of available users to log in as. This setting can be toggled to disable showing the user list.

- Type: dconf
- Key: /org/gnome/login-screen/disable-user-list
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Avoid showing user list</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\login-screen\disable-user-list</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Show notification banners

Whether notification banners are visible for application notifications.

- Type: dconf
- Key: /org/gnome/desktop/notifications/show-banners
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Show notification banners</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\desktop\notifications\show-banners</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

Show notifications in the lock screen

Whether notifications are shown in the lock screen or not.

- Type: dconf
- Key: /org/gnome/desktop/notifications/show-in-lock-screen
- Default: true
Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Show notifications in the lock screen</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\desktop\notifications\show-in-lock-screen</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

**Enable Toolkit Accessibility**

Whether toolkits should load accessibility related modules.

- Type: dconf
- Key: /org/gnome/desktop/interface/toolkit-accessibility
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Computer Policies -&gt; Ubuntu -&gt; Login Screen -&gt; Enable Toolkit Accessibility</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\gdm\dconf\org\gnome\desktop\interface\toolkit-accessibility</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>Machine</td>
</tr>
</tbody>
</table>

### 3.7.2 User Policies

**Ubuntu**

**Desktop**

**Accessibility**

**On-screen keyboard**

Whether the on-screen keyboard is turned on.

- Type: dconf
- Key: /org/gnome/desktop/11y/applications/screen-keyboard-enabled
- Default: false
Active Directory GPO support

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Accessibility -&gt; On-screen keyboard</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\a11y\applications\screen-keyboard-enabled</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Screen magnifier

Whether the screen magnifier is turned on.

- Type: dconf
  - Key: /org/gnome/desktop/a11y/applications/screen-magnifier-enabled
  - Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Accessibility -&gt; Screen magnifier</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\a11y\applications\screen-magnifier-enabled</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Screen reader

Whether the screen reader is turned on.

- Type: dconf
  - Key: /org/gnome/desktop/a11y/applications/screen-reader-enabled
  - Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Accessibility -&gt; Screen reader</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\a11y\applications\screen-reader-enabled</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>
**Enable Toolkit Accessibility**

Whether toolkits should load accessibility related modules.
- **Type**: dconf
- **Key**: /org/gnome/desktop/interface/toolkit-accessibility
- **Default**: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Accessibility -&gt; Enable Toolkit Accessibility</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\interface\toolkit-accessibility</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

**Background**

**Picture Options**

Determines how the image set by wallpaper_filename is rendered. Possible values are “none”, “wallpaper”, “centered”, “scaled”, “stretched”, “zoom”, “spanned”.
- **Type**: dconf
- **Key**: /org/gnome/desktop/background/picture-options
- **Default**: ‘zoom’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Valid values**

- none
- wallpaper
- centered
- scaled
- stretched
- zoom
- spanned

**Metadata**
Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Background -&gt; Picture Options</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\background\picture-options</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Picture URI (dark)

URI to use for the background image. Note that the backend only supports local (file://) URIs.

- Type: dconf
- Key: /org/gnome/desktop/background/picture-uri-dark
- Default for 22.04: ‘file:///usr/share/backgrounds/warty-final-ubuntu.png’
- Default for 23.04: ‘file:///usr/share/backgrounds/warty-final-ubuntu.png’
- Default for 23.10: ‘file:///usr/share/backgrounds/ubuntu-wallpaper-d.png’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 22.04, 23.04, 23.10.

#### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Background -&gt; Picture URI (dark)</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\background\picture-uri-dark</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Picture URI

URI to use for the background image. Note that the backend only supports local (file://) URIs.

- Type: dconf
- Key: /org/gnome/desktop/background/picture-uri
- Default: ‘file:///usr/share/backgrounds/warty-final-ubuntu.png’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

#### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Background -&gt; Picture URI</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\background\picture-uri</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>
**Keyboard shortcuts**

**Launch settings**

Binding to launch GNOME Settings.

- **Type:** dconf
- **Key:** /org/gnome/settings-daemon/plugins/media-keys/control-center
- **Default:** [' ']

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Keyboard shortcuts -&gt; Launch settings</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\media-keys\control-center</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

**Modifier to use for extended window management operations**

This key will initiate the “overlay”, which is a combination window overview and application launching system. The default is intended to be the “Windows key” on PC hardware. It’s expected that this binding either the default or set to the empty string.

- **Type:** dconf
- **Key:** /org/gnome/mutter/overlay-key
- **Default:** ‘Super_L’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Keyboard shortcuts -&gt; Modifier to use for extended window management operations</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\mutter\overlay-key</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>
Show the activities overview

- Type: dconf
- Key: /org/gnome/desktop/wm/keybindings/panel-main-menu
- Default: ['<Alt>F1']

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Keyboard shortcuts -&gt; Show the activities overview</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\wm\keybindings\panel-main-menu</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Launch terminal

Binding to launch the terminal.

- Type: dconf
- Key: /org/gnome/settings-daemon/plugins/media-keys/terminal
- Default: ['<Primary><Alt>t']

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Keyboard shortcuts -&gt; Launch terminal</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\settings-daemon\plugins\media-keys\terminal</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Keybinding to open the “Show Applications” view

Keybinding to open the “Show Applications” view of the Activities Overview.

- Type: dconf
- Key: /org/gnome/shell/keybindings/toggle-application-view
- Default: ['"<Super>a"']

Note: default system value is used for “Not Configured” and enforced if “Disabled”.
Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata
Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Keyboard shortcuts -&gt; Keybinding to open the “Show Applications” view</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\shell\keybindings\toggle-application-view</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Keybinding to open the overview

Keybinding to open the Activities Overview.

- Type: dconf
- Key: /org/gnome/shell/keybindings/toggle-overview
- Default for 20.04: ["<Super>s"]
- Default for 22.04: ["<Super>s"]
- Default for 23.04: ["<Super>s"]
- Default for 23.10: []

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Keyboard shortcuts -&gt; Keybinding to open the overview</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\shell\keybindings\toggle-overview</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Screensaver

### Disable lock screen

Prevent the user to lock his screen.

- Type: dconf
- Key: /org/gnome/desktop/lockdown/disable-lock-screen
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata
Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Screensaver -&gt; Disable lock screen</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-lock-screen</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class</td>
<td>User</td>
</tr>
</tbody>
</table>

**Picture Options**

Determines how the image set by wallpaper_filename is rendered. Possible values are “none”, “wallpaper”, “centered”, “scaled”, “stretched”, “zoom”, “spanned”.

- Type: dconf
- Key: /org/gnome/desktop/screensaver/picture-options
- Default: ‘zoom’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Valid values**

- none
- wallpaper
- centered
- scaled
- stretched
- zoom
- spanned

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Screensaver -&gt; Picture Options</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\screensaver\picture-options</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class</td>
<td>User</td>
</tr>
</tbody>
</table>

**Picture URI**

URI to use for the background image. Note that the backend only supports local (file://) URIs.

- Type: dconf
- Key: /org/gnome/desktop/screensaver/picture-uri
- Default: ‘file:///usr/share/backgrounds/warty-final-ubuntu.png’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.
Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Screensaver -&gt; Picture URI</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\screensaver\picture-uri</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Show notifications in the lock screen

Whether notifications are shown in the lock screen or not.

- Type: dconf
- Key: /org/gnome/desktop/notifications/show-in-lock-screen
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Screensaver -&gt; Show notifications in the lock screen</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\notifications\show-in-lock-screen</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Shell

Clock

Whether the clock displays in 24h or 12h format

Whether the clock displays in 24h or 12h format

- Type: dconf
- Key: /org/gnome/desktop/interface/clock-format
- Default: ‘24h’

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Valid values

- 24h
- 12h
Active Directory GPO support

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; Clock -&gt; Whether the clock displays in 24h or 12h format</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\interface\clock-format</td>
</tr>
<tr>
<td>Element type</td>
<td>dropdownList</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Show date in clock

If true, display date in the clock, in addition to time.

- Type: dconf
  - Key: /org/gnome/desktop/interface/clock-show-date
  - Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; Clock -&gt; Show date in clock</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\interface\clock-show-date</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Show weekday in clock

If true, display weekday in the clock, in addition to time.

- Type: dconf
  - Key: /org/gnome/desktop/interface/clock-show-weekday
  - Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; Clock -&gt; Show weekday in clock</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\interface\clock-show-weekday</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>
**LockDown**

**Disable command line**

Prevent the user from accessing the terminal or specifying a command line to be executed. For example, this would disable access to the panel’s “Run Application” dialog.

- Type: dconf
- Key: /org/gnome/desktop/lockdown/disable-command-line
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable command line</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-command-line</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

**Disable log out**

Prevent the user from logging out.

- Type: dconf
- Key: /org/gnome/desktop/lockdown/disable-log-out
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable log out</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-log-out</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>
**Active Directory GPO support**

### Disable print setup

Prevent the user from modifying print settings. For example, this would disable access to all applications’ “Print Setup” dialogs.

- **Type:** dconf
- **Key:** /org/gnome/desktop/lockdown/disable-print-setup
- **Default:** false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable print setup</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-print-setup</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Disable printing

Prevent the user from printing. For example, this would disable access to all applications’ “Print” dialogs.

- **Type:** dconf
- **Key:** /org/gnome/desktop/lockdown/disable-printing
- **Default:** false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable printing</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-printing</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Disable saving files to disk

Prevent the user from saving files to disk. For example, this would disable access to all applications’ “Save as” dialogs.

- **Type:** dconf
- **Key:** /org/gnome/desktop/lockdown/disable-save-to-disk
- **Default:** false
Active Directory GPO support

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable saving files to disk</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-save-to-disk</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Disable user switching

Prevent the user from switching to another account while his session is active.

- Type: dconf
- Key: /org/gnome/desktop/lockdown/disable-user-switching
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable user switching</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\disable-user-switching</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Disable user administration

Stop the user from modifying user accounts. By default, we allow adding and removing users, as well as changing other users settings.

- Type: dconf
- Key: /org/gnome/desktop/lockdown/user-administration-disabled
- Default: false

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata
### Active Directory GPO support

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; LockDown -&gt; Disable user administration</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\lockdown\user-administration-disabled</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Notifications

#### Show notification banners

Whether notification banners are visible for application notifications.

- Type: dconf
- Key: /org/gnome/desktop/notifications/show-banners
- Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; Notifications -&gt; Show notification banners</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\notifications\show-banners</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### List of desktop file IDs for favorite applications

The applications corresponding to these identifiers will be displayed in the favorites area.

- Type: dconf
- Key: /org/gnome/shell/favorite-apps
Active Directory GPO support


Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; List of desktop file IDs for favorite applications</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\shell\favorite-apps</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Have file manager handle the desktop

If set to true, then file manager will draw the icons on the desktop.

• Type: dconf
• Key: /org/gnome/desktop/background/show-desktop-icons
• Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Desktop -&gt; Shell -&gt; Have file manager handle the desktop</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\background\show-desktop-icons</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

Show applications button

Show applications button in the dash

• Type: dconf
• Key: /org/gnome/shell/extensions/dash-to-dock/show-show-apps-button
• Default: true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

Metadata
### Peripherals

#### Whether to automatically mount media

If set to true, then Nautilus will automatically mount media such as user-visible hard disks and removable media on start-up and media insertion.

- **Type:** dconf
- **Key:** /org/gnome/desktop/media-handling/automount
- **Default:** true

Note: default system value is used for “Not Configured” and enforced if “Disabled”.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

### Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Peripherals -&gt; Whether to automatically mount media</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\dconf\org\gnome\desktop\media-handling\automount</td>
</tr>
<tr>
<td>Element type</td>
<td>boolean</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

### Session management

#### User Drive Mapping

**User mounts**

Define network shares that will be mounted for the client. If more shares are defined higher the GPO hierarchy, the entries listed here will be appended to the list and duplicates will be removed.

Values should be in the format: :/// e.g. nfs://example_nfs.com/nfs_shared_dir smb://example_smb.com/smb_shared_dir ftp://ftp_share_server.com

This pattern must be followed, otherwise the policy will not be applied.

By the default, the mounts will be done in anonymous mode. In case of authentication needed, a krb5 tag can be added to the value, e.g. [krb5]:///

If the tag is added, the mount will require Kerberos authentication in order to occur.

The supported protocols are the same as the ones supported by gvfs. They are listed on the man page of gvfs, under the gvfs-backends section: [https://manpages.ubuntu.com/manpages/jammy/en/man7/gvfs.7.html](https://manpages.ubuntu.com/manpages/jammy/en/man7/gvfs.7.html) It’s up to the user to
ensure that the requested protocols are valid and supported and that the shared directories have the correct configuration for the requested connection.

- **Type:** mount
- **Key:** /user-mounts

**Note:**
- **Enabled:** The value(s) referenced in the entry are applied on the client machine.
- **Disabled:** The value(s) are removed from the target machine.
- **Not configured:** Value(s) declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Session management -&gt; User Drive Mapping -&gt; User mounts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\mount\user-mounts</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

**User Scripts**

**Logoff scripts**

Define scripts that are executed when the user exits from last session. Those scripts are ordered, one by line, and relative to SYSVOL/ubuntu/scripts/ directory. Scripts from this GPO will be appended to the list of scripts referenced higher in the GPO hierarchy.

- **Type:** scripts
- **Key:** /logoff

**Note:**
- **Enabled:** The scripts in the text entry are executed at user logoff time.
- **Disabled:** The scripts will be skipped. The set of scripts are per session, and refreshed only on new session creation.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

**Metadata**

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Session management -&gt; User Scripts -&gt; Logoff scripts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\scripts\logoff</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>
Active Directory GPO support

Logon scripts

Define scripts that are executed the first time an user logon until it exits from all sessions. Those scripts are ordered, one by line, and relative to SYSVOL/ubuntu/scripts/ directory. Scripts from this GPO will be appended to the list of scripts referenced higher in the GPO hierarchy.

- Type: scripts
- Key: /logon

Note: -
- Enabled: The scripts in the text entry are executed at user logon time.
- Disabled: The scripts will be skipped. The set of scripts are per session, and refreshed only on new session creation.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Session management -&gt; User Scripts -&gt; Logon scripts</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\scripts\logon</td>
</tr>
<tr>
<td>Element type</td>
<td>multiText</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

User application confinement

AppArmor

Define an AppArmor user profile to be parsed and loaded on client machines. The profile is specified as a file path relative to the SYSVOL/ubuntu/apparmor/ directory. On the client machine, user profiles are stored in /etc/apparmor.d/adsys/users/, thus the administrator can reference abstractions and tunables shipped with the client distribution of AppArmor.

The profile will ideally contain a mapping between a user and a role. Roles must be configured beforehand in the System-wide application confinement section. Below is an example of a user profile declaration:

```plaintext
include <abstractions/authentication> include <abstractions/nameservice>
capability dac_override, capability setgid, capability setuid, /etc/default/su r, /etc/environment r, @{HOME-DIRS}/.xauth* w, /usr/bin/{,b,d,rb}ash Px -> default_user, /usr/bin/{c,k,tc}sh Px -> default_user,
```

The GPO client will wrap this into an apparmor block declaration containing the client username. The default_user role must be declared beforehand in the Machine section. More details and examples can be found in the apparmor section of the adsys documentation.

The configured profile will override any profile referenced higher in the GPO hierarchy.

- Type: apparmor
- Key: /apparmor-users

Note: -
- Enabled: The profile in the text entry is applied on the client machine.
• Disabled: The profile is removed from the target machine, and any related rules are unloaded.

• Not configured: A profile declared higher in the GPO hierarchy will be used if available.

Supported on Ubuntu 20.04, 22.04, 23.04, 23.10.

An Ubuntu Pro subscription on the client is required to apply this policy.

Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>User Policies -&gt; Ubuntu -&gt; Session management -&gt; User application confinement -&gt; AppArmor</td>
</tr>
<tr>
<td>Registry Key</td>
<td>Software\Policies\Ubuntu\apparmor\apparmor-users</td>
</tr>
<tr>
<td>Element type</td>
<td>text</td>
</tr>
<tr>
<td>Class:</td>
<td>User</td>
</tr>
</tbody>
</table>

3.8 Supported releases

ADSys is supported on Ubuntu starting from **20.04.2 LTS**, and tested with Windows Server 2019.

Only Active Directory on-premise is supported.

3.9 Recommended readings

• adsysd help or man adsysd.

• adsysctl help or man adsysctl.
Discussion and clarification of key topics.

4.1 Managers

ADSys supports a wide variety of managers to configure and control various aspects of the client systems. Managers available with ADSys are

4.1.1 Dconf manager

The dconf manager allows to enforce default or custom dconf settings on the client. Some settings are globally enforced on the machine while other are per-user specific settings.

Example of settings

- Change the login screen layout and background color.
- Set and lock the default applications in the launcher.
- Set and lock the user wallpaper.
- Set the date and time format of the clock.

Rules precedence

Any settings will override the same settings in less specific GPO.

Settings UI

Widgets

Depending on the type of settings, appropriate widgets are displayed to the AD system administrator. Those can be different per release if the type of settings changed.
Active Directory GPO support

States

Enabled

Setting a key to enabled will apply a value to any machines or user targeted by the GPO. It allows the Active Directory administrator to enter a value that will be applied to the target object (user or machine). This setting will be enforced on the client. Only an administrator of the client system can override it, but it will be reset to the Active Directory setting on next refresh.
Disabled

Setting a key to `disabled` will prevent user updates. However, no value can be explicitly entered by the Active Directory administrator. The default value of the client system will then be used (which may differ between machines).
Not configured

Finally, not configured is the default state. The setting is managed as usual directly on the client and without Active Directory.

4.1.2 Admin privileges management

The Admin privilege manager allows to grant or revoke superuser privileges for the default local user, and Active Directory users and groups.

All those settings are globally enforced on the machine and are available at Computer Configuration > Policies > Administrative Templates > Ubuntu > Client management > Privilege Authorization.
Feature availability

This feature is available only for subscribers of **Ubuntu Pro**.
Active Directory GPO support

Rules precedence

Any settings will override the same settings in less specific GPO.

What does administrator means?

Administrators:

- Can get administrators privileges and ran commands as such with sudo.
- Are considered admin for all polkit actions. If the current user is not an admin and a particular daemon require polkit administrator privilege, a prompt will allow you to choose an existing administrators to authenticate before performing the action.

Local user

Members of the local sudo group are administrators by default on the machine.

Not Configured or enabled

This status keep the default for the system: sudo group members are considered administrators on the client.

Disabled

sudo group members are not considered administrators on the client.

Note: you can grant specific users not necessarily in the sudo group administrator privileges with the “Client administrator option”.

Active Directory users and groups

Users and groups in the directory can be granted administrator privileges of the local machine with sudo.

Several users or groups or a set of both can be assigned.

The form is a list of users and group, one per line, user@domain for a user and %group@domain for a group.

Not Configured or disabled

There is no AD user or group configured with admin privileges for the machine.

Enabled

There is one or several AD user or group configured with admin privileges for the machine via the list under it.

Note: you can use this list to grant non-default local users matching the name on the client.
4.1.3 Scripts execution

The scripts managers allows AD administrators to target scripts to be executed on behalf of the client, or by users. Those scripts, can be triggered on:

- Computer startup and shutdown. They are located in Computer Configuration > Policies > Administrative Templates > Ubuntu > Client management > Computer Scripts.
- User log on and log off. They are located in User Configuration > Policies > Administrative Templates > Ubuntu > Session management > User Scripts.

Scripts can be shell scripts, or any binary that can be executed on Linux.

Feature availability

This feature is available only for subscribers of Ubuntu Pro.
Active Directory GPO support

Rules precedence

Any settings will be additive to the same settings in less specific GPO. It means that scripts in the less specific GPO will be executed first.

Installing scripts on sysvol

Scripts must be available in the assets sharing directory on your Active Directory sysvol/ samba share.

In this directory, next to Policies in your domain folder, create a directory matching your distribution name. For instance Ubuntu, which will be the assets sharing directory.

It must also contain a GPT.ini file of the form:

```
[General]
Version=22
displayName=Ubuntu Assets Directory
```

Every time you change the scripts, you need to increase the version stanza in the GPT.ini file (similarly to how Active Directory is doing automatically when you change any field). This will signal clients that a new version of assets (including scripts) are available and should be downloaded.

Then, place any scripts you need under the scripts/ directory (subdirectories are allowed).
Active Directory GPO support

Automating the incrementation of the GPT.ini version stanza

Making manual changes to a file every time scripts are changed can be unproductive and tedious. For your convenience, we developed a tool to automate this process. For detailed usage and installation instructions please refer to the *Active Directory Watch Daemon* documentation.

Active directory UI

Enabled

The form is a list of scripts path, relative to the `scripts/` subdirectory of your asset sharing file system, one per line.
Active Directory GPO support

Not configured or Disabled

This GPO won’t refer any scripts for execution.

Scripts behaviors

Scripts erroring out

If a script errors out on execution, it will not fail the session startup or the machine boot. However, some errors details will be available in systemd journal.

Incorrect script path reference

If a script referenced by a GPO doesn’t exist or that the path is incorrect, then the policy will fail to be applied and any client startup or user log on will fail.

Transactional sessions

Scripts sessions are transitional: if you installed V1 of some scripts, and starts a session (computer startup or user log on), then you can be ensured that whatever version is updated on the Active Directory, you will exit the session with the same V1 version of the scripts you initially provided (computer log off or user log off).

However, even if user1 has logged on with version V1 of the scripts and V2 is available, then any log on for user2 will use the V2 of the scripts. user1 though, is ensured to continue using the V1 version of the scripts.

4.1.4 Managing AppArmor profiles

The AppArmor manager allows to enforce custom AppArmor profiles on the client.

Custom AppArmor profiles can be enforced on a:

- System-wide level, located in Computer Configuration > Policies > Administrative Templates > Ubuntu > Client management > System-wide application confinement > AppArmor
- User level, located in Computer Configuration > Policies > Administrative Templates > Ubuntu > Session management > User application confinement > AppArmor

Feature availability

This feature is available only for subscribers of Ubuntu Pro.
Rules precedence

On a system-wide level, files in the entry are appended to the list of profiles referenced higher in the GPO hierarchy. On a user level, the configured profile will override any profile referenced higher in the GPO hierarchy.

Installing AppArmor profiles on sysvol

AppArmor profiles must be available in the assets sharing directory on your Active Directory sysvol/samba share. In this directory, next to Policies in your domain folder, create a directory matching your distribution name. For instance Ubuntu, which will be the assets sharing directory.

It must also contain a GPT.ini file of the form:

```ini
[General]
Version=22
displayName=Ubuntu Assets Directory
```

Every time you change the contents of AppArmor profiles, you need to increase the version stanza in the GPT.ini file (similarly to how Active Directory is doing automatically when you change any field). This will signal clients that a new version of assets (including AppArmor profiles) are available and should be downloaded.

Then, place any AppArmor profiles you need under the apparmor/ directory (subdirectories are allowed).
Active Directory GPO support

Automating the incrementation of the GPT.ini version stanza

Making manual changes to a file every time AppArmor profiles are changed can be unproductive and tedious. For your convenience, we developed a tool to automate this process. For detailed usage and installation instructions please refer to the *Active Directory Watch Daemon* documentation.

Developing AppArmor profiles

We highly recommend developing AppArmor profiles separately from ADSys, testing them, and only then integrating them with ADSys. Here are some resources to get you started:

- How to create AppArmor Profile
- AppArmor Documentation
- The Comprehensive Guide To AppArmor

System-wide profiles

The form is a list of AppArmor profile paths, relative to the `apparmor/` subdirectory of your assets sharing file system, one per line.
Active Directory GPO support

When set to enabled, adsys will load the configured AppArmor profiles on refresh. AppArmor's caching functionality is leveraged to ensure redundant reloads are kept to a minimum, i.e. a loaded profile will be parsed again only if a change occurred in the profile definition.

On the client machine, system-wide profiles are located under `/etc/apparmor.d/adsys/machine` by default.

When set disabled / not configured, ADSys will unload any previously loaded profiles (that were managed by ADSys) from the client machine.
User profiles

AppArmor supports confining executables on a user-by-user basis via the **pam_apparmor** PAM module. The module allows applications to confine authenticated users into subprofiles based on group names, user names, or a default profile. To accomplish this, **pam_apparmor** needs to be registered as a PAM session module. A working example can be found on the official AppArmor repository wiki.

The form accepts a path to a single file, relative to the **apparmor/** directory of your assets sharing file system.
Installing the AppArmor PAM module

The PAM module can be installed on Ubuntu using the following command:

```
sudo apt install libpam-apparmor
```

The module must then be configured manually for any desired executables. To enable it for the `su` command, append the following to the `/etc/pam.d/su` file:

```
session optional pam_apparmor.so order=user,default
```

**Warning** Even though GPOs can be applied to AD groups, the AppArmor policy manager currently only supports confining AD users, so we’ve omitted `group` from the PAM order.

User profile declaration syntax

As per the example linked above, the user profile is essentially a subprofile (also known as a hat) which needs to be included in the actual executable profile definition. The PAM module will then attempt to change hats into a subprofile containing the target user’s name.

If a regular subprofile looks like the following (assuming `admin@domain.com` is the user we want to confine):

```
^admin@domain.com {
    #include <abstractions/authentication>
    #include <abstractions/nameservice>
    capability dac_override,
    capability setgid,
    capability setuid,
    /etc/default/su r,
    /etc/environment r,
    @{HOMEDIRS}/.xauth* w,
    /bin/{,b,d,rb}ash Ux,
    /bin/{c,k,tc}sh Ux,
}
```

Its ADSys counterpart will omit the first and last lines which will be inferred automatically when policies are applied on the client machine. Thus, the subprofile declaration becomes:

```
#include <abstractions/authentication>
#include <abstractions/nameservice>
capability dac_override,
capability setgid,
capability setuid,
/etc/default/su r,
/etc/environment r,
@{HOMEDIRS}/.xauth* w,
/bin/{,b,d,rb}ash Ux,
/bin/{c,k,tc}sh Ux,
```

When the policy is applied on the target machine, the user profile will be created in `/etc/apparmor.d/adsys/users/admin@domain.com`.

The user profile doesn’t accomplish anything on its own until it is included in a system-wide profile definition. We recommend including the entire `users` directory in the profile declaration, as evidenced by the official `pam_apparmor` documentation.
Active Directory GPO support

It is also recommended to define a DEFAULT subprofile as part of the system-wide profiles to ensure `pam_apparmor` has a profile to switch to as a last resort if it cannot find a profile for the user.

```
/usr/bin/su {
  ...
  include "users"

  ^DEFAULT {
    capability dac_override,
    capability setgid,
    capability setuid,
    /etc/default/su r,
    /etc/environment r,
    @{HOMEDIRS}/.xauth* w,
    /usr/bin/{,b,d,rb}ash Px -> default_user,
    /usr/bin/{c,k,tc}sh Px -> default_user,
  }
}
```

Profile parsing behaviors

Troubleshooting misbehaving user profiles

If you encounter issues with user profiles, it's always helpful to check the kernel buffer (`dmesg`) and the authentication logs at `/var/log/auth.log`. Additionally, appending `debug` to the PAM module declaration stanzas at `/etc/pam.d/` will log additional debug information, such as the profile that the module tries to switch to.

```
  administrator@warthogs.biz(uid=1130200500) by root(uid=0)
  'administrator@warthogs.biz'
```

In contrast, switching to a user with no declared profile and no default profile will print the following and restrict access entirely:

```
Nov 18 14:00:22 ubuntu2204 su: pam_unix(su:session): session opened for user
  fry@warthogs.biz(uid=1130201105) by root(uid=0)
Nov 18 14:00:22 ubuntu2204 su: pam_apparmor(su:session): Using username 'fry@warthogs.biz'
Nov 18 14:00:22 ubuntu2204 su: pam_apparmor(su:session): Can't change to any hat
Nov 18 14:00:22 ubuntu2204 su: pam_unix(su:session): session closed for user
  fry@warthogs.biz
```

The PAM module goes through the order specified in the configuration (user, default), ultimately bailing and denying access if it cannot find any hat to switch to.
Active Directory GPO support

**Error on loading profiles**

ADSys relies on the `apparmor_parser` executable to parse, load, and unload profiles. If the command fails for any reason (e.g. syntax errors in profile declaration), loading profiles will be aborted and the output of the `apparmor_parser` command will be logged.

```
ERROR Error from server: error while updating policy: failed to apply policy to 
"ubuntu2204": can't apply apparmor policy to ubuntu2204: can't apply machine policy:
failed to get apparmor policies: exit status 1
AppArmor parser error for /etc/apparmor.d/adsys/machine/pam_roles in profile /etc/
apparmor.d/adsys/machine/pam_roles at line 9: Lexer found unexpected character: '<'
(0x3c) in state: INITIAL
```

**Invalid profile path reference**

If an AppArmor profile referenced by a GPO doesn’t exist or the path is incorrect, then the policy will fail to be applied and any client startup or user log on will fail.

**4.1.5 Network shares**

The mount managers allow AD administrators to specify network shares that must be mounted in the file system when a client logs in.

**Feature Availability**

This feature is available only for subscribers of **Ubuntu Pro**.

**System mounts**

The mount process for these mounts is triggered at the moment a client logs in. System mounts are handled by systemd through unit files and happen at root level. Therefore, users do not have control over the mounting/unmounting process.

All protocols supported by the `mount` command should work out of the box. However, the only tested ones are `smb`, `ftp` and `nfs`.

The backends for the protocols `smb` and `nfs` are automatically enabled when installing the `adsys` package. In order to enable the backend for `ftp` mounts, the user must install the recommended `curlftpfs` package. This behavior is tested on Ubuntu and might differ on other Linux distributions.

Access control and file permissions should be configured on the shared location.

User mount policies are located under **Computer Configuration > Policies > Administrative Templates > Ubuntu > Client management > System Drive Mapping**, as shown in the following picture.
Setting up the policy

The form is a list of shared drives that should be mounted for the client machine. They must follow the structure `{protocol}://[host name or ip address]/[shared location]`.

The default mount behavior is to mount the listed shares anonymously. In order to require kerberos authentication for the mount process, the tag `[krb5]` can be added as a prefix to the listed share, i.e. `[krb5]{protocol}://[host name or ip address]/[shared location]`.

Additional mount options are not supported yet.

All entries must be separated by a line break.
The policy strategy is “append”. Therefore, if multiple policies defining network shares are to be applied to a client, all of the listed shares will be mounted.

Duplicated shares will be handled. Anonymous and authenticated shares of the same location are treated as duplicates and the first one listed will take precedence over the others.

Rules precedence
Active Directory GPO support

Invalid mounts

ADSys will block client authentication only if the policy cannot be applied, meaning that the listed shares could not be set up. Any issues that arise after the setup process, such as an unreachable domain or a non-existent share, will be reported as an error. Refer to the system logs for more details about the failures.

Unmounting

The unmounting process is handled by systemd on shutdown.

User mounts

The mount process for these mounts is triggered at the moment a user logs in. User mounts are accessible in the file manager and the user has the ability to unmount them manually.

Credentials authentication for mounts are disabled on ADSys. Instead, authentication is done with the Kerberos ticket present on the machine. If the mount is set to anonymous, then the administrator must ensure that the shared drive supports anonymous access and that the permissions for the directory are set accordingly.

User mount policies are located under User Configuration > Policies > Administrative Templates > Ubuntu > Session management > User Drive Mapping, as shown in the following picture.
Setting up the policy

The format is a list of shared drives that should be mounted for the user. They must follow the structure `{protocol}://[{host name or ip address}]/[{shared location}]`. If the drive is to be mounted anonymously, the tag `[anonymous]` should be added as a prefix to the listed entry, i.e. `[anonymous]{protocol}://[{host name or ip address}]/[{shared location}]`.

All entries must be separated by a line break.

The mount process is handled with GVfs and it defines in which directory the shared drive will be mounted into. Usually, it’s mounted under `/run/user/%U/gvfs/`. 
Active Directory GPO support

Rules precedence

The policy strategy is “append”. Therefore, if multiple policies defining mount locations are to be applied to a user, all of the listed entries will be mounted.

Invalid mounts

If the mounting of an entry listed in the policy fails, ADSys will proceed with the other entries in the policy, mounting those it can and logging those that cannot be mounted.

Unmounting

The unmounting process is handled by systemd at the end of the session.

4.1.6 Network Proxy

The proxy manager allows AD administrators to apply proxy settings on the clients. Currently, only system-wide proxy settings are supported.

Proxy settings are configurable under the following GPO path:

- System-wide level, located in Computer Configuration > Policies > Administrative Templates > Ubuntu > Client management > System proxy configuration
Feature availability

This feature is available only for subscribers of **Ubuntu Pro**.

Additionally, the `ubuntu-proxy-manager` package must be installed in order for proxy settings to be applied on the client system. On Ubuntu systems, run the following to install the package:

```
sudo apt install ubuntu-proxy-manager
```
Active Directory GPO support

Rules precedence

Configured proxy settings will override any settings referenced higher in the GPO hierarchy.

Setting up the policy

The System proxy configuration category provides a list of configurable proxy settings:
- HTTP Proxy
- HTTPS Proxy
- FTP Proxy
- SOCKS Proxy
- Ignored hosts
- Auto configuration URL
Configured settings will then be forwarded to `ubuntu-proxy-manager` which will apply them on all supported backends (e.g. environment variables, APT, GSettings). For an up-to-date list of supported backends, proxy formats and behaviors, refer to the `ubuntu-proxy-manager` documentation.

**Disabling proxy settings**

To disable or remove proxy settings, either set the required values to an empty value (""), or mark the setting as `Disabled`.

Note that if none of the proxy settings in the category are set (all settings are `Not Configured`), the proxy manager won’t take any action.

**Troubleshooting manager errors**

If any proxy GPOs are configured and the `ubuntu-proxy-manager` package is not installed (specifically, no response is received from the D-Bus call on the object exported by the proxy manager service), the manager will fail hard. The package doesn’t need to be installed if no proxy entries are configured.

If proxy application fails for other reasons, refer to the Troubleshooting section of the `ubuntu-proxy-manager` documentation for details on how to debug the D-Bus service.

### 4.1.7 Certificate auto-enrollment

The certificate policy manager allows clients to enroll for certificates from **Active Directory Certificate Services**. Certificates are then continuously monitored and refreshed by the `certmonger` daemon. Currently, only machine certificates are supported.

Unlike the other ADSys policy managers which are configured in the special Ubuntu section provided by the ADMX files (Administrative Templates), settings for certificate auto-enrollment are configured in the Microsoft GPO tree:

Active Directory GPO support

Feature availability

This feature is available only for subscribers of Ubuntu Pro and has been tested and known to work on all Ubuntu versions starting with 22.04 (Jammy).

Additionally, the following packages must be installed on the client in order for auto-enrollment to work:

- certmonger - daemon that monitors and updates certificates
- cepces - certmonger extension that can communicate with Active Directory Certificate Services

On Ubuntu systems, run the following to install them:

```bash
sudo apt install certmonger python3-cepces
```

On the Windows side, the following roles must be installed and configured:

- Certification Authority
- Certificate Enrollment Policy Web Service
- Certificate Enrollment Web Service
Active Directory GPO support

- Network Device Enrollment Service

**Rules precedence**

Auto-enrollment configuration will override any settings referenced higher in the GPO hierarchy.

**Policy configuration**

Certificate auto-enrollment is configured by setting the **Configuration Model** to **Enabled** and ticking the following checkbox: **Update certificates that use certificate templates**.
Active Directory GPO support

The policy can be disabled by performing *any* of the following:

- unticking the **Update certificates that use certificate templates** checkbox
- setting the **Configuration Model** to **Disabled** or **Not configured**

The other settings in this GPO entry do not affect ADSys in any way.

For more advanced configuration, a list of policy servers can be specified in the following GPO entry:

Applying the policy

On the client system, a successful auto-enrollment will place certificate data in the following paths:

- `/var/lib/adsys/certs` - certificate data
- `/var/lib/adsys/private/certs` - private key data
- `/usr/local/share/ca-certificates` - root certificate data (symbolic link pointing to `/var/lib/adsys/certs`)

For detailed information on the tracked certificates, `certmonger` can be directly interacted with:

```
# Query monitored certificates
> getcert list
Number of certificates and requests being tracked: 1.
Request ID 'galacticcafe-CA.Machine':
  status: MONITORING
  stuck: no
  key pair storage: type=FILE,location='/var/lib/adsys/private/certs/galacticcafe-CA.Machine.key'
  certificate: type=FILE,location='/var/lib/adsys/certs/galacticcafe-CA.Machine.crt'
  CA: galacticcafe-CA
  issuer: CN=galacticcafe-CA,DC=galacticcafe,DC=com
  subject: CN=keypress.galacticcafe.com
  issued: 2023-08-18 18:44:27 EEST
  expires: 2024-08-17 18:44:27 EEST
  dns: keypress.galacticcafe.com
  key usage: digitalSignature,keyEncipherment
  eku: id-kp-clientAuth,id-kp-serverAuth
  certificate template/profile: Machine
  profile: Machine
  pre-save command:
  post-save command:
  track: yes
  auto-renew: yes

# Query known CAs
> getcert list-cas
(...)
CA 'galacticcafe-CA':
  is-default: no
  ca-type: EXTERNAL
  helper-location: /usr/libexec/certmonger/cepces-submit --server=win-mk85nrq26nu.galacticcafe.com --auth=Kerberos
```
Active Directory GPO support

Policy implementation

With the exception of policy parsing, ADSys leverages the Samba implementation of certificate auto-enrollment. As this feature is only available in newer versions of Samba, we have taken the liberty of vendoring the required Samba files to allow this policy to work on Ubuntu versions that ship an older Samba version. These files are shipped in /usr/share/adsys/python/vendor_samba.

To ensure idempotency when applying the policy, we set up a Samba TDB cache file at /var/lib/adsys/samba/cert_gpo_state_$(hostname).tdb which contains various information pertaining to the enrolled certificate(s).

Here is an overview of what happens during policy application:

- GPO parsing (ADSys)
- execute Python helper script (ADSys)
- fetch root CA and policy servers (Samba)
- start monitoring certificate using certmonger and cepces (Samba)

Troubleshooting

Some dependencies are not available in the client Ubuntu installation

While certmonger has been available for a while in Ubuntu, python3-cepces is a new package, available starting with Ubuntu 23.10. If unavailable on the client version, it can also be manually installed from the source repository. The certificate policy manager only checks for the existence of the cepces-submit and getcert binaries, not their respective packages, in order to allow some wiggle room for this.

Manipulating certificates with getcert

While not encouraged, certificates can be manipulated with the same tool. This could be helpful for debugging purposes.

```bash
# Regenerate a certificate
> getcert rekey -i galacticcafe-CA.Machine
Resubmitting "galacticcafe-CA.Machine" to "galacticcafe-CA".

# Unmonitor a certificate
> getcert stop-tracking -i galacticcafe-CA.Machine
Request "galacticcafe-CA.Machine" removed.

# Remove CA
> getcert remove-ca -c galacticcafe-CA
CA "galacticcafe-CA" removed.
```

Note that tampering with certificate data outside of ADSys (e.g. manually unmonitoring using getcert) will render the GPO cache obsolete as it will cause a drift between the actual state and the “known” cached state. In this case, it’s best to remove the cache file at /var/lib/adsys/samba/*.tdb together with any enrolled certificates and CAs to ensure a clean slate.
Active Directory GPO support

Debugging auto-enroll script

While certificate parsing happens in ADSys itself, enrollment is done via an embedded Python helper script. For debugging purposes, it can be dumped to the current directory and made executable by executing the following commands:

```
> adsysctl policy debug cert-autoenroll-script
> chmod +x ./cert-autoenroll
```

Before executing the script manually, the following environment variables have to be set:

```
export PYTHONPATH=/usr/share/adsys/python
export KRB5CCNAME=/var/run/adsys/krb5cc/$(hostname)
```

Then, run the script passing the required arguments (the argument list is also printed in the AD Sys debug logs during policy application):

```
# Un-enroll machine
> ./cert-autoenroll unenroll keypress galacticcafe.com --state_dir /var/lib/adsys --debug
```

Errors communicating with the CEP/CES servers

If AD Sys successfully applies the policy but `getcert list` does not list the certificates or they are in an unexpected state, check the `certmonger` logs for details (`journalctl -u certmonger`). Additionally, debug logging for `cepces` can be enabled by editing the logging configuration at `/etc/cepces/logging.conf`.

The `cepces` configuration itself is batteries-included, meaning it should work out of the box for most setups. All configuration options are documented and configurable at `/etc/cepces/cepces.conf`.

Additional information

While configuring Active Directory Certificate Services is outside the scope of the policy manager documentation, we have found the following resources to be useful:

- How to setup Microsoft Active Directory Certificate Services
- How to increase your CSR key size on Microsoft IIS without removing the production certificate?

Acknowledgements

We would like to thank the Samba team for making great strides in the research and implementation of certificate auto-enrollment via Active Directory Certificate Services.

4.1.8 Case of the security policy

Certain group policies are directly managed by SSSD. In such instances, AD Sys is not involved at all. This is applicable to Security Settings.

In Windows Group Policy Management Editor, you can locate these keys at `[FOREST.ROOT] > Computer Configuration > Windows Settings > Security Settings`

Below is a table providing a non-comprehensive list of Security Settings defined in Windows, which are not managed by AD Sys but receive partial support through SSSD.
### Windows Setting

<table>
<thead>
<tr>
<th><strong>Account Policies &gt; Password Policy</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Enforce password history</td>
</tr>
<tr>
<td>Maximum password age</td>
</tr>
<tr>
<td>Minimum password age</td>
</tr>
<tr>
<td>Minimum password length</td>
</tr>
<tr>
<td>Password must meet complexity requirements</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Account Policies &gt; Account Lockout Policy</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Account lockout duration</td>
</tr>
<tr>
<td>Account lockout threshold</td>
</tr>
<tr>
<td>Reset account lockout counter after</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Local Policies &gt; User Rights Assignment</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Access this computer from the network</td>
</tr>
<tr>
<td>Allow log on locally</td>
</tr>
<tr>
<td>Allow log on through Remote Desktop Services</td>
</tr>
<tr>
<td>Change the system time</td>
</tr>
<tr>
<td>Change the timezone</td>
</tr>
<tr>
<td>Deny access to this computer from the network</td>
</tr>
<tr>
<td>Deny log on as a batch job</td>
</tr>
<tr>
<td>Deny log on as a service</td>
</tr>
<tr>
<td>Deny log on locally</td>
</tr>
<tr>
<td>Deny log on through Remote Desktop Services</td>
</tr>
<tr>
<td>Log on as a batch job</td>
</tr>
<tr>
<td>Log on as a service</td>
</tr>
<tr>
<td>Shutdown the system</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Local Policies / Security Options</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator account status</td>
</tr>
<tr>
<td>Shutdown: Allow system to be shut down without having to log on</td>
</tr>
</tbody>
</table>

Get more information on SSSD.
IN THIS DOCUMENTATION

Tutorials

Start here: a hands-on introduction to ADSys for new users

How-to guides

Step-by-step guides covering key operations and common tasks

Explanation

Discussion and clarification of key topics

Reference

Technical information - specifications, APIs, architecture
PROJECT AND COMMUNITY

ADSys is a member of the Ubuntu family. It’s an open source project that warmly welcomes community contributions, suggestions, fixes and constructive feedback.

- Code of conduct
- Join us in the Ubuntu Community
- Contribute or Report an issue
- Thinking about using ADSys for your next project? Get in touch!
- Licensed under GPL v3