autosuspend

Release 2.0.7.dev0

Jan 18, 2020

Contents:

1 Installation instructions **3**
1.1 Requirements . . . 3
1.2 Binary packages . . . 3
1.3 From-source installation . . . 4

2 Configuration file **5**
2.1 Syntax . . . 5
2.2 General configuration . . . 6
2.3 Activity check configuration . . . 6
2.4 Wake up check configuration . . . 7

3 Available activity checks **9**
3.1 ActiveCalendarEvent . . . 9
3.2 ActiveConnection . . . 10
3.3 ExternalCommand . . . 10
3.4 Kodi . . . 10
3.5 KodiIdleTime . . . 11
3.6 Load . . . 11
3.7 LogindSessionsIdle . . . 12
3.8 Mpd . . . 12
3.9 NetworkBandwidth . . . 12
3.10 Ping . . . 13
3.11 Processes . . . 13
3.12 Smb . . . 13
3.13 Users . . . 14
3.14 XIdleTime . . . 14
3.15 XPath . . . 15

4 Available wake up checks **17**
4.1 Calendar . . . 17
4.2 Command . . . 18
4.3 File . . . 18
4.4 Periodic . . . 18
4.5 XPath . . . 19
4.6 XPathDelta . . . 19

5 systemd integration **21**

5.1 Preventing the system from sleeping immediately after waking up 21

6 On-demand wakeup 23

7 Debugging 25

8 Manpages 27
8.1 autosuspend 27
8.2 autosuspend.conf 28

9 Changelog 41
9.1 2.0.6 41
9.2 2.0.5 41
9.3 2.0.4 41
9.4 2.0.3 41
9.5 2.0.2 42
9.6 2.0 42

10 Support 45

11 Indices and tables 47

Index 49

`autosuspend` is a daemon that periodically suspends a system on inactivity and wakes it up again automatically in case it is needed. For this purpose, `autosuspend` periodically iterates a number of user-configurable activity checks, which indicate whether an activity on the host is currently present that should prevent the host from suspending. In case one of the checks indicates such activity, no action is taken and periodic checking continues. Otherwise, in case no activity can be detected, this state needs to be present for a specified amount of time before the host is suspended by `autosuspend`. In addition to the activity checks, wake up checks are used to determine planned future activities of the system (for instance, a TV recording or a periodic backup). In case such activities are known before suspending, `autosuspend` triggers a command to wake up the system automatically before the soonest activity.

CHAPTER 1

Installation instructions

autosuspend is designed for Python **3** and does not work with Python 2.

1.1 Requirements

The minimal requirements are.

- Python 3
- psutil

Additionally, the some checks need further dependencies to function properly. Please refer to *Available activity checks* for individual requirements.

If checks using URLs to load data should support `file://` URLs, requests-file is needed.

1.2 Binary packages

1.2.1 Debian (testing)

Installation from official package sources:

```
apt-get install autosuspend
```

1.2.2 Archlinux (AUR)

autosuspend is available as an Archlinux AUR package.

Installation via **pacaur**:

```
pacaur -S autosuspend
```

1.2.3 Other distributions

In case you want to generate a package for a different Linux distribution, I'd be glad to hear about that.

1.3 From-source installation

autosuspend provides a usual `setup.py` file for installation using common setuptools methods. Briefly, the following steps are necessary to install **autosuspend**:

```
git clone https://github.com/languitar/autosuspend.git
cd autosuspend
python3 setup.py install # with desired options
```

To build the documentation, the following command can be used:

```
python3 setup.py build_sphinx
```

CHAPTER 2

Configuration file

2.1 Syntax

The **autosuspend** configuration file uses INI syntax and needs to be processable by the Python configparser module.

A simple configuration file could look like:

```
[general]
interval = 30
idle_time = 900
suspend_cmd = /usr/bin/systemctl suspend
wakeup_cmd = echo {timestamp:.0f} > /sys/class/rtc/rtc0/wakealarm
notify_cmd_wakeup = su myuser -c notify-send -a autosuspend 'Suspending the system.␣
↪Wake up at {iso}'
notify_cmd_no_wakeup = su myuser -c notify-send -a autosuspend 'Suspending the system.
↪'

[check.Ping]
enabled = false
hosts = 192.168.0.7

[check.RemoteUsers]
class = Users
enabled = true
name = .*
terminal = .*
host = [0-9].*

[wakeup.File]
enabled = True
path = /var/run/autosuspend/wakeup
```

The configuration file consists of a `[general]` section, which specifies general processing options, and multiple sections of the format `[check.*]` and `[wakeup.*]`. These sections describe the activity and wake up checks to execute.

2.2 General configuration

The `[general]` section contains options controlling the overall behavior of the **autosuspend** daemon. These are:

interval
: The time to wait after executing all checks in seconds.

idle_time
: The required amount of time in seconds with no detected activity before the host will be suspended. Default: 300 seconds

min_sleep_time
: The minimal amount of time in seconds the system has to sleep for actually triggering suspension. If a scheduled wake up results in an effective time below this value, the system will not sleep. Default: 1200 seconds

wakeup_delta
: Wake up the system this amount of seconds earlier than the time that was determined for an event that requires the system to be up. This value adds a safety margin for the time a the wake up effectively takes. Default: 30 seconds

suspend_cmd
: The command to execute in case the host shall be suspended. This line can contain additional command line arguments to the command to execute.

wakeup_cmd
: The command to execute for scheduling a wake up of the system. The given string is processed using Python's `str.format()` and a format argument called `timestamp` encodes the UTC timestamp of the planned wake up time (float). Additionally `iso` can be used to acquire the timestamp in ISO 8601 format.

notify_cmd_wakeup
: A command to execute before the system is going to suspend for the purpose of notifying interested clients. This command is only called in case a wake up is scheduled. The given string is processed using Python's `str.format()` and a format argument called `timestamp` encodes the UTC timestamp of the planned wake up time (float). Additionally `iso` can be used to acquire the timestamp in ISO 8601 format. If empty or not specified, no command will be called.

notify_cmd_no_wakeup
: A command to execute before the system is going to suspend for the purpose of notifying interested clients. This command is only called in case NO wake up is scheduled. Hence, no string formatting options are available. If empty or not specified, no command will be called.

woke_up_file
: Location of a file that indicates to **autosuspend** that the computer has suspended since the last time checks were executed. This file is usually created by a systemd service. Thus, changing the location also requires adapting the respective service. Refer to *systemd integration* for further details.

2.3 Activity check configuration

For each activity check to execute, a section with the name format `[check.*]` needs to be created. Each check has a name and an executing class which implements the behavior. The fraction of the section name `check.` determines the name, and in case no class option is given inside the section, also the class which implements the check. In case the *`class`* option is specified, the name is completely user-defined and the same check can even be instantiated multiple times with differing names.

For each check, these generic options can be specified:

`class`
: Name of the class implementing the check. If the name does not contain a dot (`.`), this is assumed to be one of the checks provided by autosuspend internally. Otherwise, this can be used to pull in third-party checks. If this option is not specified, the section name must represent a valid internal check class.

`enabled`
: Needs to be `true` for a check to actually execute. `false` is assumed if not specified.

Furthermore, each check might have custom options.

2.4 Wake up check configuration

Wake up checks uses the same configuration logic as the previously described activity checks. However, the configuration file sections start with `wakeup.` instead of `check.`.

For options of individual checks, please refer to *Available activity checks* and *Available wake up checks*.

CHAPTER 3

Available activity checks

The following checks for activity are currently implemented. Each of the is described with its available configuration options and required optional dependencies.

3.1 ActiveCalendarEvent

Checks an online iCalendar file for events that are currently running. If so, this indicates activity and prevents suspending the system. Thus, a calendar can be provided with times at which the system should not go to sleep. If this calendar resides on an online service like a groupware it might even be possible to invite the system.

3.1.1 Options

`url`
The URL to query for the iCalendar file

`timeout`
Timeout for executed requests in seconds. Default: 5.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

3.1.2 Requirements

- requests
- icalendar

- dateutil
- tzlocal

3.2 ActiveConnection

Checks whether there is currently a client connected to a TCP server at certain ports. Can be used to e.g. block suspending the system in case SSH users are connected or a web server is used by clients.

3.2.1 Options

`ports`
list of comma-separated port numbers

3.2.2 Requirements

3.3 ExternalCommand

Executes an arbitrary command. In case this command returns 0, the system is assumed to be active.

The command is executed as is using shell execution. Beware of malicious commands in obtained configuration files.

3.3.1 Options

`command`
The command to execute including all arguments

3.3.2 Requirements

3.4 Kodi

Checks whether an instance of Kodi is currently playing.

3.4.1 Options

`url`
Base URL of the JSON RPC API of the Kodi instance, default: `http://localhost:8080/jsonrpc`

`timeout`
Request timeout in seconds, default: `5`

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

3.4.2 Requirements

- requests

3.5 KodiIdleTime

Checks whether there has been interaction with the Kodi user interface recently. This prevents suspending the system in case someone is currently browsing collections etc. This check is redundant to `XIdleTime` on systems using an X server, but might be necessary in case Kodi is used standalone. It does not replace the `Kodi` check, as the idle time is not updated when media is playing.

3.5.1 Options

`idle_time`
Marks the system active in case a user interaction has appeared within the this amount of seconds until now. Default: `120`

`url`
Base URL of the JSON RPC API of the Kodi instance, default: `http://localhost:8080/jsonrpc`

`timeout`
Request timeout in seconds, default: `5`

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

3.5.2 Requirements

- requests

3.6 Load

Checks whether the system load 5 is below a certain value.

3.6.1 Options

`threshold`
a float for the maximum allowed load value, default: 2.5

3.6.2 Requirements

3.7 LogindSessionsIdle

Prevents suspending in case `IdleHint` for one of the running sessions logind sessions is set to `no`. Support for setting this hint currently varies greatly across display managers, screen lockers etc. Thus, check exactly whether the hint is set on your system via `loginctl show-session`.

3.7.1 Options

`types`
A comma-separated list of sessions types to inspect for activity. The check ignores sessions of other types. Default: `tty`, `x11`, `wayland`

`states`
A comma-separated list of session states to inspect. For instance, `lingering` sessions used for background programs might not be of interest. Default: `active`, `online`

3.7.2 Requirements

- dbus-python

3.8 Mpd

Checks whether an instance of MPD is currently playing music.

3.8.1 Options

`host`
Host containing the MPD daemon, default: `localhost`

`port`
Port to connect to the MPD daemon, default: `6600`

`timeout`
Request timeout in seconds, default: `5`

3.8.2 Requirements

- python-mpd2

3.9 NetworkBandwidth

Checks whether more network bandwidth is currently being used than specified. A set of specified interfaces is checked in this regard, each of the individually, based on the average bandwidth on that interface. This average is based on the global checking interval specified in the configuration file via the *`interval`* option.

3.9.1 Options

`interfaces`
Comma-separated list of network interfaces to check

`threshold_send` `<byte/s>`
If the average sending bandwidth of one of the specified interfaces is above this threshold, then activity is detected. Specified in bytes/s, default: `100`

`threshold_receive` `<byte/s>`
If the average receive bandwidth of one of the specified interfaces is above this threshold, then activity is detected. Specified in bytes/s, default: `100`

3.9.2 Requirements

3.10 Ping

Checks whether one or more hosts answer to ICMP requests.

3.10.1 Options

`hosts`
Comma-separated list of host names or IPs.

3.10.2 Requirements

3.11 Processes

If currently running processes match an expression, the suspend will be blocked. You might use this to hinder the system from suspending when for example your rsync runs.

3.11.1 Options

`processes`
list of comma-separated process names to check for

3.11.2 Requirements

3.12 Smb

Any active Samba connection will block suspend.

3.12.1 Options

`smbstatus`
executable needs to be present.

3.12.2 Requirements

3.13 Users

Checks whether a user currently logged in at the system matches several criteria. All provided criteria must match to indicate activity on the host.

3.13.1 Options

All regular expressions are applied against the full string. Capturing substrings needs to be explicitly enabled using wildcard matching.

`name`
A regular expression specifying which users to capture, default: `.*`.

`terminal`
A regular expression specifying the terminal on which the user needs to be logged in, default: `.*`.

`host`
A regular expression specifying the host from which a user needs to be logged in, default: `.*`.

3.13.2 Requirements

3.14 XIdleTime

Checks whether all active local X displays have been idle for a sufficiently long time. Determining which X11 sessions currently exist on a running system is a harder problem than one might expect. Sometimes, the server runs as root, sometimes under the real user, and many other configuration variants exist. Thus, multiple sources for active X serer instances are implemented for this check, each of them having different requirements and limitations. They can be changed using the provided configuration option.

3.14.1 Options

`timeout`
required idle time in seconds

`method`
The method to use for acquiring running X sessions. Valid options are `sockets` and `logind`. The default is `sockets`.

`sockets` Uses the X server sockets files found in `/tmp/.X11-unix`. This method requires that all X server instances run with user permissions and not as root.

`logind` Uses logind to obtain the running X server instances. This does not support manually started servers.

`ignore_if_process`
A regular expression to match against the process names executed by each X session owner. In case the use has a running process that matches this expression, the X idle time is ignored and the check continues as if there was no activity. This can be useful in case of processes which inevitably tinker with the idle time.

`ignore_users`
Do not check sessions of users matching this regular expressions.

3.14.2 Requirements

- dbus-python for the `logind` method

3.15 XPath

A generic check which queries a configured URL and expects the reply to contain XML data. The returned XML document is checked against a configured XPath expression and in case the expression matches, the system is assumed to be active.

Some common applications and their respective configuration are:

tvheadend The required URL for tvheadend is (if running on the same host):

```
http://127.0.0.1:9981/status.xml
```

In case you want to prevent suspending in case there are active subscriptions or recordings, use the following XPath:

```
/currentload/subscriptions[number(.) > 0] | /currentload/recordings/recording/
↪start
```

If you have a permantently running subscriber like Kodi, increase the `0` to `1`.

Plex For Plex, use the following URL (if running on the same host):

```
http://127.0.0.1:32400/status/sessions/?X-Plex-Token={TOKEN}
```

Where acquiring the token is documented here.

If suspending should be prevented in case of any activity, this simple XPath expression will suffice:

```
/MediaContainer[@size > 2]
```

3.15.1 Options

`url`
The URL to query for the XML reply.

`xpath`
The XPath query to execute. In case it returns a result, the system is assumed to be active.

`timeout`
Timeout for executed requests in seconds. Default: 5.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

3.15.2 Requirements

- requests
- lxml

CHAPTER 4

Available wake up checks

The following checks for wake up times are currently implemented. Each of the checks is described with its available configuration options and required optional dependencies.

4.1 Calendar

Determines next wake up time from an iCalendar file. The next event that starts after the current time is chosen as the next wake up time.

Remember that updates to the calendar can only be reflected in case the system currently running. Changes to the calendar made while the system is sleeping will obviously not trigger an earlier wake up.

4.1.1 Options

`url`
The URL to query for the XML reply.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

`xpath`
The XPath query to execute. Must always return number strings or nothing.

`timeout`
Timeout for executed requests in seconds. Default: 5.

4.1.2 Requirements

- requests
- icalendar
- dateutil
- tzlocal

4.2 Command

Determines the wake up time by calling an external command The command always has to succeed. If something is printed on stdout by the command, this has to be the next wake up time in UTC seconds.

The command is executed as is using shell execution. Beware of malicious commands in obtained configuration files.

4.2.1 Options

`command`
The command to execute including all arguments

4.3 File

Determines the wake up time by reading a file from a configured location. The file has to contains the planned wake up time as an int or float in seconds UTC.

4.3.1 Options

`path`
path of the file to read in case it is present

4.4 Periodic

Always schedules a wake up at a specified delta from now on. Can be used to let the system wake up every once in a while, for instance, to refresh the calendar used in the `Calendar` check.

4.4.1 Options

`unit`
A string indicating in which unit the delta is specified. Valid options are: `microseconds`, `milliseconds`, `seconds`, `minutes`, `hours`, `days`, `weeks`.

`value`
The value of the delta as an int.

4.5 XPath

A generic check which queries a configured URL and expects the reply to contain XML data. The returned XML document is parsed using a configured XPath expression that has to return timestamps UTC (as strings, not elements). These are interpreted as the wake up times. In case multiple entries exist, the soonest one is used.

4.5.1 Options

`url`
The URL to query for the XML reply.

`xpath`
The XPath query to execute. Must always return number strings or nothing.

`timeout`
Timeout for executed requests in seconds. Default: 5.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

4.6 XPathDelta

Comparable to `XPath`, but expects that the returned results represent the wake up time as a delta to the current time in a configurable unit.

This check can for instance be used for tvheadend with the following expression:

```
//recording/next/text()
```

4.6.1 Options

`url`
The URL to query for the XML reply.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

`xpath`
The XPath query to execute. Must always return number strings or nothing.

`timeout`
Timeout for executed requests in seconds. Default: 5.

`unit`

A string indicating in which unit the delta is specified. Valid options are: `microseconds`, `milliseconds`, `seconds`, `minutes`, `hours`, `days`, `weeks`. Default: minutes

CHAPTER 5

systemd integration

Even though it is possible to run **autosuspend** manually (cf. *the manpage*), in production use cases, the daemon will usually be run from systemd. For this purpose, the package ships with service definition files for systemd, so that you should be able to manage **autosuspend** via systemd. These files need to be installed in the appropriate locations for such service files, which depend on the Linux distribution. Some common locations are: * `/usr/lib/systemd/system` (e.g. Archlinux packaged service files) * `/lib/systemd/system` (e.g. Debian packaged service files) * `/etc/systemd/system` (e.g. Archlinux manually added service files) Binary installation packages for Linux distributions should have installed the service files at the appropriate locations already.

To start **autosuspend** via systemd, execute:

```
systemctl enable autosuspend.service
```

To start **autosuspend** automatically at system start, execute:

```
systemctl start autosuspend.service
```

5.1 Preventing the system from sleeping immediately after waking up

Unfortunately, **autosuspend** does not detect automatically if the system was placed into suspend mode manually. Therefore, it might happen that after waking up again, all checks have indicated inactivity for a long time (the whole phase of sleeping) and **autosuspend** might initiate suspending again immediately. To prevent this, systemd needs to inform **autosuspend** every time the system suspends. This is achieved by a seconds service file, which needs to be enabled (not started):

```
systemctl enable autosuspend-detect-suspend.service
```

CHAPTER 6

On-demand wakeup

autosuspend itself only handles wake ups for events that were foreseeable at the time the system was put into sleep mode. In case the system also has to be used on-demand, a simple way to wake up the system is to enable Wake on LAN. Here, a special network packet can be used to wake up the system again. Multiple front-ends exist to send these magic packets. The typical usage scenario with this approach is to manually send the magic packet when the system is needed, wait a few seconds, and then to perform the intended tasks with the system.

Wake on LAN needs to be specifically enabled on the system. Typically, the documentation of common Linux distributions explains how to enable Wake on LAN:

- Archlinux
- Debian
- Ubuntu

A set of front-ends for various platforms allows to send the magic packets. For instance:

- gWakeOnLan: GTK GUI, Linux
- wol: command line, Linux
- Wake On Lan: GUI, Windows
- Wake On Lan: Android
- Wake On Lan: Android, open-source
- Kore (Kodi remote control): Android, for Kodi users
- Mocha WOL: iOS

CHAPTER 7

Debugging

In case you need to track configuration issues to understand why a system suspends or does not, the extensive logging output of **`autosuspend`** might be used. The command line flag *`autosuspend -l`* allows to specify a Python logging configuration file which specifies what to log. The provided systemd service files (see *systemd integration*) already use `/etc/autosuspend-logging.conf` as the standard location and a default file is usually installed. Each iteration logs exactly which condition detected activity or not. So you should be able to find out what is going on.

In case one of the conditions you monitor prevents suspending the system if an external connection is established (logged-in users, open TCP port), then the logging configuration file can be changed to use the broadcast-logging package. This way, the server will broadcast new log messages on the network and external clients on the same network can listen to these messages without creating an explicit connection. Please refer to the documentation of the broadcast-logging package on how to enable and use it. Additionally, one might also examine the `journalctl` for **`autosuspend`** after the fact.

CHAPTER 8

Manpages

8.1 autosuspend

8.1.1 Synopsis

autosuspend [*options*]

8.1.2 Description

`autosuspend` is a daemon that periodically suspends a system on inactivity and wakes it up again automatically in case it is needed. For this purpose, **`autosuspend`** periodically iterates a number of user-configurable activity checks, which indicate whether an activity on the host is currently present that should prevent the host from suspending. In case one of the checks indicates such activity, no action is taken and periodic checking continues. Otherwise, in case no activity can be detected, this state needs to be present for a specified amount of time before the host is suspended by **`autosuspend`**. In addition to the activity checks, wake up checks are used to determine planned future activities of the system (for instance, a TV recording or a periodic backup). In case such activities are known before suspending, **`autosuspend`** triggers a command to wake up the system automatically before the soonest activity.

If not specified via a command line argument, **`autosuspend`** looks for a default configuration at `/etc/autosuspend.conf`. *`autosuspend.conf(5)`* describes the configuration file, the available checks, and their configuration options.

8.1.3 Options

`-h, --help`
: Displays an online help.

`-c` `FILE`, **`--config`** `FILE`
: Specifies an alternate config file to use instead of the default on at `/etc/autosuspend.conf`.

-a, --allchecks
Usually, **autosuspend** stops checks in each iteration as soon as the first matching check indicates system activity. If this flag is set, all subsequent checks are still executed. Useful mostly for debugging purposes.

-r SECONDS, **--runfor** SECONDS
If specified, do not run endlessly. Instead, operate only for the specified amount of seconds, then exit. Useful mostly for debugging purposes.

-l [FILE], **--logging** [FILE]
If used without a file argument, enable debug logging (use as last argument). If used with a file, configure logging with the provided logging file. This file needs to follow the conventions for Python logging files.

8.1.4 Bugs

Please report bugs at the project repository at https://github.com/languitar/autosuspend.

8.1.5 See also

autosuspend.conf(5)

8.2 autosuspend.conf

8.2.1 Synopsis

/etc/autosuspend.conf

8.2.2 Description

Configures the **autosuspend** daemon.

Syntax

The **autosuspend** configuration file uses INI syntax and needs to be processable by the Python configparser module.

A simple configuration file could look like:

```
[general]
interval = 30
idle_time = 900
suspend_cmd = /usr/bin/systemctl suspend
wakeup_cmd = echo {timestamp:.0f} > /sys/class/rtc/rtc0/wakealarm
notify_cmd_wakeup = su myuser -c notify-send -a autosuspend 'Suspending the system.␣
↪Wake up at {iso}'
notify_cmd_no_wakeup = su myuser -c notify-send -a autosuspend 'Suspending the system.
↪'

[check.Ping]
enabled = false
hosts = 192.168.0.7

[check.RemoteUsers]
```

(continues on next page)

(continued from previous page)

```
class = Users
enabled = true
name = .*
terminal = .*
host = [0-9].*

[wakeup.File]
enabled = True
path = /var/run/autosuspend/wakeup
```

The configuration file consists of a `[general]` section, which specifies general processing options, and multiple sections of the format `[check.*]` and `[wakeup.*]`. These sections describe the activity and wake up checks to execute.

General configuration

The `[general]` section contains options controlling the overall behavior of the **`autosuspend`** daemon. These are:

`interval`
: The time to wait after executing all checks in seconds.

`idle_time`
: The required amount of time in seconds with no detected activity before the host will be suspended. Default: 300 seconds

`min_sleep_time`
: The minimal amount of time in seconds the system has to sleep for actually triggering suspension. If a scheduled wake up results in an effective time below this value, the system will not sleep. Default: 1200 seconds

`wakeup_delta`
: Wake up the system this amount of seconds earlier than the time that was determined for an event that requires the system to be up. This value adds a safety margin for the time a the wake up effectively takes. Default: 30 seconds

`suspend_cmd`
: The command to execute in case the host shall be suspended. This line can contain additional command line arguments to the command to execute.

`wakeup_cmd`
: The command to execute for scheduling a wake up of the system. The given string is processed using Python's `str.format()` and a format argument called `timestamp` encodes the UTC timestamp of the planned wake up time (float). Additionally `iso` can be used to acquire the timestamp in ISO 8601 format.

`notify_cmd_wakeup`
: A command to execute before the system is going to suspend for the purpose of notifying interested clients. This command is only called in case a wake up is scheduled. The given string is processed using Python's `str.format()` and a format argument called `timestamp` encodes the UTC timestamp of the planned wake up time (float). Additionally `iso` can be used to acquire the timestamp in ISO 8601 format. If empty or not specified, no command will be called.

`notify_cmd_no_wakeup`
: A command to execute before the system is going to suspend for the purpose of notifying interested clients. This command is only called in case NO wake up is scheduled. Hence, no string formatting options are available. If empty or not specified, no command will be called.

woke_up_file
: Location of a file that indicates to **autosuspend** that the computer has suspended since the last time checks were executed. This file is usually created by a systemd service. Thus, changing the location also requires adapting the respective service. Refer to *systemd integration* for further details.

Activity check configuration

For each activity check to execute, a section with the name format `[check.*]` needs to be created. Each check has a name and an executing class which implements the behavior. The fraction of the section name `check.` determines the name, and in case no class option is given inside the section, also the class which implements the check. In case the *`class`* option is specified, the name is completely user-defined and the same check can even be instantiated multiple times with differing names.

For each check, these generic options can be specified:

class
: Name of the class implementing the check. If the name does not contain a dot (`.`), this is assumed to be one of the checks provided by autosuspend internally. Otherwise, this can be used to pull in third-party checks. If this option is not specified, the section name must represent a valid internal check class.

enabled
: Needs to be `true` for a check to actually execute. `false` is assumed if not specified.

Furthermore, each check might have custom options.

Wake up check configuration

Wake up checks uses the same configuration logic as the previously described activity checks. However, the configuration file sections start with `wakeup.` instead of `check.`.

The options of individual checks are outlined below.

8.2.3 Available activity checks

The following checks for activity are currently implemented. Each of the is described with its available configuration options and required optional dependencies.

ActiveCalendarEvent

Checks an online iCalendar file for events that are currently running. If so, this indicates activity and prevents suspending the system. Thus, a calendar can be provided with times at which the system should not go to sleep. If this calendar resides on an online service like a groupware it might even be possible to invite the system.

Options

url
: The URL to query for the iCalendar file

timeout
: Timeout for executed requests in seconds. Default: 5.

username
: Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

Requirements

- requests
- icalendar
- dateutil
- tzlocal

ActiveConnection

Checks whether there is currently a client connected to a TCP server at certain ports. Can be used to e.g. block suspending the system in case SSH users are connected or a web server is used by clients.

Options

`ports`
list of comma-separated port numbers

Requirements

ExternalCommand

Executes an arbitrary command. In case this command returns 0, the system is assumed to be active.

The command is executed as is using shell execution. Beware of malicious commands in obtained configuration files.

Options

`command`
The command to execute including all arguments

Requirements

Kodi

Checks whether an instance of Kodi is currently playing.

Options

`url`
Base URL of the JSON RPC API of the Kodi instance, default: `http://localhost:8080/jsonrpc`

`timeout`
Request timeout in seconds, default: `5`

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

Requirements

- requests

KodiIdleTime

Checks whether there has been interaction with the Kodi user interface recently. This prevents suspending the system in case someone is currently browsing collections etc. This check is redundant to `XIdleTime` on systems using an X server, but might be necessary in case Kodi is used standalone. It does not replace the `Kodi` check, as the idle time is not updated when media is playing.

Options

`idle_time`
Marks the system active in case a user interaction has appeared within the this amount of seconds until now. Default: `120`

`url`
Base URL of the JSON RPC API of the Kodi instance, default: `http://localhost:8080/jsonrpc`

`timeout`
Request timeout in seconds, default: `5`

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

Requirements

- requests

Load

Checks whether the system load 5 is below a certain value.

Options

`threshold`
a float for the maximum allowed load value, default: 2.5

Requirements

LogindSessionsIdle

Prevents suspending in case `IdleHint` for one of the running sessions logind sessions is set to `no`. Support for setting this hint currently varies greatly across display managers, screen lockers etc. Thus, check exactly whether the hint is set on your system via `loginctl show-session`.

Options

`types`
: A comma-separated list of sessions types to inspect for activity. The check ignores sessions of other types. Default: `tty`, `x11`, `wayland`

`states`
: A comma-separated list of session states to inspect. For instance, `lingering` sessions used for background programs might not be of interest. Default: `active`, `online`

Requirements

- dbus-python

Mpd

Checks whether an instance of MPD is currently playing music.

Options

`host`
: Host containing the MPD daemon, default: `localhost`

`port`
: Port to connect to the MPD daemon, default: `6600`

`timeout`
: Request timeout in seconds, default: `5`

Requirements

- python-mpd2

NetworkBandwidth

Checks whether more network bandwidth is currently being used than specified. A set of specified interfaces is checked in this regard, each of the individually, based on the average bandwidth on that interface. This average is based on the global checking interval specified in the configuration file via the *`interval`* option.

Options

`interfaces`
Comma-separated list of network interfaces to check

`threshold_send` `<byte/s>`
If the average sending bandwidth of one of the specified interfaces is above this threshold, then activity is detected. Specified in bytes/s, default: `100`

`threshold_receive` `<byte/s>`
If the average receive bandwidth of one of the specified interfaces is above this threshold, then activity is detected. Specified in bytes/s, default: `100`

Requirements

Ping

Checks whether one or more hosts answer to ICMP requests.

Options

`hosts`
Comma-separated list of host names or IPs.

Requirements

Processes

If currently running processes match an expression, the suspend will be blocked. You might use this to hinder the system from suspending when for example your rsync runs.

Options

`processes`
list of comma-separated process names to check for

Requirements

Smb

Any active Samba connection will block suspend.

Options

`smbstatus`
executable needs to be present.

Requirements

Users

Checks whether a user currently logged in at the system matches several criteria. All provided criteria must match to indicate activity on the host.

Options

All regular expressions are applied against the full string. Capturing substrings needs to be explicitly enabled using wildcard matching.

`name`
A regular expression specifying which users to capture, default: `.*`.

`terminal`
A regular expression specifying the terminal on which the user needs to be logged in, default: `.*`.

`host`
A regular expression specifying the host from which a user needs to be logged in, default: `.*`.

Requirements

XIdleTime

Checks whether all active local X displays have been idle for a sufficiently long time. Determining which X11 sessions currently exist on a running system is a harder problem than one might expect. Sometimes, the server runs as root, sometimes under the real user, and many other configuration variants exist. Thus, multiple sources for active X serer instances are implemented for this check, each of them having different requirements and limitations. They can be changed using the provided configuration option.

Options

`timeout`
required idle time in seconds

`method`
The method to use for acquiring running X sessions. Valid options are `sockets` and `logind`. The default is `sockets`.

`sockets` Uses the X server sockets files found in `/tmp/.X11-unix`. This method requires that all X server instances run with user permissions and not as root.

`logind` Uses logind to obtain the running X server instances. This does not support manually started servers.

`ignore_if_process`
A regular expression to match against the process names executed by each X session owner. In case the use has a running process that matches this expression, the X idle time is ignored and the check continues as if there was no activity. This can be useful in case of processes which inevitably tinker with the idle time.

`ignore_users`
Do not check sessions of users matching this regular expressions.

Requirements

- dbus-python for the `logind` method

XPath

A generic check which queries a configured URL and expects the reply to contain XML data. The returned XML document is checked against a configured XPath expression and in case the expression matches, the system is assumed to be active.

Some common applications and their respective configuration are:

tvheadend The required URL for tvheadend is (if running on the same host):

```
http://127.0.0.1:9981/status.xml
```

In case you want to prevent suspending in case there are active subscriptions or recordings, use the following XPath:

```
/currentload/subscriptions[number(.) > 0] | /currentload/recordings/recording/
↪start
```

If you have a permantently running subscriber like Kodi, increase the `0` to `1`.

Plex For Plex, use the following URL (if running on the same host):

```
http://127.0.0.1:32400/status/sessions/?X-Plex-Token={TOKEN}
```

Where acquiring the token is documented here.

If suspending should be prevented in case of any activity, this simple XPath expression will suffice:

```
/MediaContainer[@size > 2]
```

Options

`url`
The URL to query for the XML reply.

`xpath`
The XPath query to execute. In case it returns a result, the system is assumed to be active.

`timeout`
Timeout for executed requests in seconds. Default: 5.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

Requirements

- requests
- lxml

8.2.4 Available wake up checks

The following checks for wake up times are currently implemented. Each of the checks is described with its available configuration options and required optional dependencies.

Calendar

Determines next wake up time from an iCalendar file. The next event that starts after the current time is chosen as the next wake up time.

Remember that updates to the calendar can only be reflected in case the system currently running. Changes to the calendar made while the system is sleeping will obviously not trigger an earlier wake up.

Options

`url`
The URL to query for the XML reply.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

`password`
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

`xpath`
The XPath query to execute. Must always return number strings or nothing.

`timeout`
Timeout for executed requests in seconds. Default: 5.

Requirements

- requests
- icalendar
- dateutil
- tzlocal

Command

Determines the wake up time by calling an external command The command always has to succeed. If something is printed on stdout by the command, this has to be the next wake up time in UTC seconds.

The command is executed as is using shell execution. Beware of malicious commands in obtained configuration files.

Options

`command`
The command to execute including all arguments

File

Determines the wake up time by reading a file from a configured location. The file has to contains the planned wake up time as an int or float in seconds UTC.

Options

`path`
path of the file to read in case it is present

Periodic

Always schedules a wake up at a specified delta from now on. Can be used to let the system wake up every once in a while, for instance, to refresh the calendar used in the `Calendar` check.

Options

`unit`
A string indicating in which unit the delta is specified. Valid options are: `microseconds`, `milliseconds`, `seconds`, `minutes`, `hours`, `days`, `weeks`.

`value`
The value of the delta as an int.

XPath

A generic check which queries a configured URL and expects the reply to contain XML data. The returned XML document is parsed using a configured XPath expression that has to return timestamps UTC (as strings, not elements). These are interpreted as the wake up times. In case multiple entries exist, the soonest one is used.

Options

`url`
The URL to query for the XML reply.

`xpath`
The XPath query to execute. Must always return number strings or nothing.

`timeout`
Timeout for executed requests in seconds. Default: 5.

`username`
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

password
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

XPathDelta

Comparable to `XPath`, but expects that the returned results represent the wake up time as a delta to the current time in a configurable unit.

This check can for instance be used for tvheadend with the following expression:

```
//recording/next/text()
```

Options

url
The URL to query for the XML reply.

username
Optional user name to use for authenticating at a server requiring authentication. If used, also a password must be provided.

password
Optional password to use for authenticating at a server requiring authentication. If used, also a user name must be provided.

xpath
The XPath query to execute. Must always return number strings or nothing.

timeout
Timeout for executed requests in seconds. Default: 5.

unit
A string indicating in which unit the delta is specified. Valid options are: `microseconds`, `milliseconds`, `seconds`, `minutes`, `hours`, `days`, `weeks`. Default: minutes

CHAPTER 9

Changelog

9.1 2.0.6

Missing changelog for 2.0.5.

9.2 2.0.5

9.2.1 Fixes bugs

- The `KodiIdleTime` checks has been fixed (#54)
- The default command for scheduling a wakeup was extended to clear a potential previous wake up time to avoid errors (#57)

9.3 2.0.4

This is a minor bug fix release.

9.3.1 Fixed bugs

- `ActiveConnection` did not handle local IPv6 addresses with scope such as `fe80::5193:518c:5c69:aedb%enp3s0` (#50)

9.4 2.0.3

This is a minor bug fix release.

9.4.1 Fixed bugs

- `NetworkBandwidth` did not update its internal state and therefore did not work as documented (#49)

9.5 2.0.2

This is a minor bug fix release.

9.5.1 Fixed bugs

- `Kodi` and `KodiIdleTime` checks now catch `JSONDecodeErrors` (#45)
- `Kodi` and `KodiIdleTime` checks now support authentication (#47)

9.6 2.0

This version adds scheduled wake ups as its main features. In addition to checks for activity, a set of checks for future activities can now be configured to determine times at which the systems needs to be online again. The daemon will start suspending in case the next detected wake up time is far enough in the future and schedule an automatic system wake up at the closest determined wake up time. This can, for instance, be used to ensure that the system is up again when a TV show has to be recorded to disk.

Below is a detailed list of notable changes.

9.6.1 New features

- Scheduled wake ups (#9).
- Ability to call configurable user commands before suspending for notification purposes (#25).
- Checks using network requests now support authentication (#32).
- Checks using network requests now support `file://` URIs (#36).

New activity checks

- `ActiveCalendarEvent`: Uses an iCalendar file (via network request) to prevent suspending in case an event in the calendar is currently active (#24).
- `KodiIdleTime`: Checks the idle time of Kodi to prevent suspending in case the menu is used (#33).

New wakeup checks

- `Calendar`: Wake up the system at the next event in an iCalendar file (requested via network, #30).
- `Command`: Call an external command to determine the next wake up time (#26).
- `File`: Read the next wake up time from a file (#9).
- `Periodic`: Wake up at a defined interval, for instance, to refresh calendars for the `Calendar` check (#34).
- `XPath` and `XPathDelta`: Request an XML document and use XPath to extract the next wakeup time.

9.6.2 Fixed bugs

- XPath checks now support responses with explicit encodings (#29).

9.6.3 Notable changes

- The namespace of the logging systems has been rearranged (#38). Existing logging configurations might require changes.
- The default configuration file has been reduced to explain the syntax and semantics. For a list of all available checks, refer the manual instead (#39).

For a complete list of all addressed issues and new features, please refer to the respective Github milestone.

CHAPTER 10

Support

In case you have found a bug or you want to request a new feature, please open an issue at the Github project.

CHAPTER 11

Indices and tables

- genindex
- search

Index

Symbols

-a, -allchecks
    autosuspend command line option, 27
-c FILE, -config FILE
    autosuspend command line option, 27
-h, -help
    autosuspend command line option, 27
-l [FILE], -logging [FILE]
    autosuspend command line option, 28
-r SECONDS, -runfor SECONDS
    autosuspend command line option, 28

A

autosuspend command line option
    -a, -allchecks, 27
    -c FILE, -config FILE, 27
    -h, -help, 27
    -l [FILE], -logging [FILE], 28
    -r SECONDS, -runfor SECONDS, 28

C

check-active-calendar-event command line option
    password, 9, 31
    timeout, 9, 30
    url, 9, 30
    username, 9, 30
check-active-connections command line option
    ports, 10, 31
check-external-command command line option
    command, 10, 31
check-kodi command line option
    password, 10, 32
    timeout, 10, 31
    url, 10, 31
    username, 10, 31
check-kodi-idle-time command line option
    idle_time, 11, 32
    password, 11, 32
    timeout, 11, 32
    url, 11, 32
    username, 11, 32
check-load command line option
    threshold, 11, 32
check-logind-idle command line option
    states, 12, 33
    types, 12, 33
check-mpd command line option
    host, 12, 33
    port, 12, 33
    timeout, 12, 33
check-network-bandwidth command line option
    interfaces, 13, 34
    threshold_receive <byte/s>, 13, 34
    threshold_send <byte/s>, 13, 34
check-ping command line option
    hosts, 13, 34
check-processes command line option
    processes, 13, 34
check-smb command line option
    smbstatus, 13, 34
check-users command line option
    host, 14, 35
    name, 14, 35
    terminal, 14, 35
check-xidletime command line option
    ignore_if_process, 14, 35
    ignore_users, 14, 35
    method, 14, 35
    timeout, 14, 35
check-xpath command line option
    password, 15, 36
    timeout, 15, 36
    url, 15, 36
    username, 15, 36
    xpath, 15, 36

class
    config-check command line option, 6, 30
command
    check-external-command command line option, 10, 31
    wakeup-command command line option, 18, 38
config-check command line option
    class, 6, 30
    enabled, 7, 30
config-general command line option
    idle_time, 6, 29
    interval, 6, 29
    min_sleep_time, 6, 29
    notify_cmd_no_wakeup, 6, 29
    notify_cmd_wakeup, 6, 29
    suspend_cmd, 6, 29
    wakeup_cmd, 6, 29
    wakeup_delta, 6, 29
    woke_up_file, 6, 29

E

enabled
    config-check command line option, 7, 30

H

host
    check-mpd command line option, 12, 33
    check-users command line option, 14, 35
hosts
    check-ping command line option, 13, 34

I

idle_time
    check-kodi-idle-time command line option, 11, 32
    config-general command line option, 6, 29
ignore_if_process
    check-xidletime command line option, 14, 35
ignore_users
    check-xidletime command line option, 14, 35
interfaces
    check-network-bandwidth command line option, 13, 34
interval
    config-general command line option, 6, 29

M

method
    check-xidletime command line option, 14, 35
min_sleep_time
    config-general command line option, 6, 29

N

name
    check-users command line option, 14, 35
notify_cmd_no_wakeup
    config-general command line option, 6, 29
notify_cmd_wakeup
    config-general command line option, 6, 29

P

password
    check-active-calendar-event command line option, 9, 31
    check-kodi command line option, 10, 32
    check-kodi-idle-time command line option, 11, 32
    check-xpath command line option, 15, 36
    wakeup-calendar command line option, 17, 37
    wakeup-xpath command line option, 19, 38
    wakeup-xpath-delta command line option, 19, 39
path
    wakeup-file command line option, 18, 38
port
    check-mpd command line option, 12, 33
ports
    check-active-connections command line option, 10, 31
processes
    check-processes command line option, 13, 34

S

smbstatus
    check-smb command line option, 13, 34
states
    check-logind-idle command line option, 12, 33
suspend_cmd

config-general command line option, 6, 29

T

terminal
check-users command line option, 14, 35

threshold
check-load command line option, 11, 32

threshold_receive <byte/s>
check-network-bandwidth command line option, 13, 34

threshold_send <byte/s>
check-network-bandwidth command line option, 13, 34

timeout
check-active-calendar-event command line option, 9, 30
check-kodi command line option, 10, 31
check-kodi-idle-time command line option, 11, 32
check-mpd command line option, 12, 33
check-xidletime command line option, 14, 35
check-xpath command line option, 15, 36
wakeup-calendar command line option, 17, 37
wakeup-xpath command line option, 19, 38
wakeup-xpath-delta command line option, 19, 39

types
check-logind-idle command line option, 12, 33

U

unit
wakeup-periodic command line option, 18, 38
wakeup-xpath-delta command line option, 19, 39

url
check-active-calendar-event command line option, 9, 30
check-kodi command line option, 10, 31
check-kodi-idle-time command line option, 11, 32
check-xpath command line option, 15, 36
wakeup-calendar command line option, 17, 37
wakeup-xpath command line option, 19, 38
wakeup-xpath-delta command line option, 19, 39

username
check-active-calendar-event command line option, 9, 30
check-kodi command line option, 10, 31
check-kodi-idle-time command line option, 11, 32
check-xpath command line option, 15, 36
wakeup-calendar command line option, 17, 37
wakeup-xpath command line option, 19, 38
wakeup-xpath-delta command line option, 19, 39

V

value
wakeup-periodic command line option, 18, 38

W

wakeup-calendar command line option
password, 17, 37
timeout, 17, 37
url, 17, 37
username, 17, 37
xpath, 17, 37

wakeup-command command line option
command, 18, 38

wakeup-file command line option
path, 18, 38

wakeup-periodic command line option
unit, 18, 38
value, 18, 38

wakeup-xpath command line option
password, 19, 38
timeout, 19, 38
url, 19, 38
username, 19, 38
xpath, 19, 38

wakeup-xpath-delta command line option
password, 19, 39
timeout, 19, 39
unit, 19, 39
url, 19, 39
username, 19, 39
xpath, 19, 39

wakeup_cmd
config-general command line option, 6, 29

wakeup_delta

config-general command line option, 6, 29

woke_up_file
    config-general command line option, 6, 29

X

xpath
    check-xpath command line option, 15, 36
    wakeup-calendar command line option, 17, 37
    wakeup-xpath command line option, 19, 38
    wakeup-xpath-delta command line option, 19, 39