```
# All configuration options described here can also be supplied on the
# command line of cups-browsed via the "-o" option. In case of
# contradicting settings the setting defined in the configuration file
# will get used.

# Unknown directives are ignored, also unknown values.


# Where should cups-browsed save information about the print queues it had
# generated when shutting down, like whether one of these queues was the
# default printer, or default option settings of the queues?

# CacheDir /var/cache/cups


# Where should cups-browsed create its debug log file (if "DebugLogging file"
# is set)?

# LogDir /var/log/cups


# How should debug logging be done? Into the file
# /var/log/cups/cups-browsed_log (’file’), to stderr (’stderr’), or
# not at all (’none’)?

# Note that if cups-browsed is running as a system service (for
# example via systemd) logging to stderr makes the log output going to
# the journal or syslog. Only if you run cups-browsed from the command
# line (for development or debugging) it will actually appear on
# stderr.

# DebugLogging file
# DebugLogging stderr
# DebugLogging file stderr
# DebugLogging none


# Which protocols will we use to discover printers on the network?
# Can use DNSSD and/or CUPS and/or LDAP, or ’none’ for neither.

BrowseRemoteProtocols dnssd cups


# Which protocols will we use to broadcast shared local printers to the network?
# Can use DNSSD and/or CUPS, or ’none’ for neither.
# Only CUPS is actually supported, as DNSSD is done by CUPS itself (we ignore
# DNSSD in this directive).

# BrowseLocalProtocols none


# Settings of this directive apply to both BrowseRemoteProtocols and
# BrowseLocalProtocols.
# Can use DNSSD and/or CUPS and/or LDAP, or ’none’ for neither.

# BrowseProtocols none


# Only browse remote printers (via DNS-SD or CUPS browsing) from
# selected servers using the "BrowseAllow", "BrowseDeny", and
# "BrowseOrder" directives

# This serves for restricting the choice of printers in print dialogs
# to trusted servers or to reduce the number of listed printers in the
# print dialogs to a more user-friendly amount in large networks with
# very many shared printers.

# This only filters the selection of remote printers for which
# cups-browsed creates local queues. If the print dialog uses other
# mechanisms to list remote printers as for example direct DNS-SD
# access, cups-browsed has no influence. cups-browsed also does not
# prevent the user from manually accessing non-listed printers.

# "BrowseAllow": Accept printers from these hosts or networks. If
# there are only "BrowseAllow" lines and no "BrowseOrder" and/or
# "BrowseDeny" lines, only servers matching at last one "BrowseAllow"
# line are accepted.

# "BrowseDeny": Deny printers from these hosts or networks. If there
# are only "BrowseDeny" lines and no "BrowseOrder" and/or
# "BrowseAllow" lines, all servers NOT matching any of the
# "BrowseDeny" lines are accepted.

# "BrowseOrder": Determine the order in which "BrowseAllow" and
# "BrowseDeny" lines are applied. With "BrowseOrder Deny,Allow" in the
# beginning all servers are accepted, then the "BrowseDeny" lines are
# applied to exclude unwished servers or networks and after that the
# "BrowseAllow" lines to re-include servers or networks. With
# "BrowseOrder Allow,Deny" we start with denying all servers, then
# applying the "BrowseAllow" lines and afterwards the "BrowseDeny"
# lines.

# Default for "BrowseOrder" is "Deny.Allow" if there are both
# "BrowseAllow" and "BrowseDeny" lines.

# If there are no "Browse..." lines at all, all servers are accepted.

# BrowseAllow All
# BrowseAllow cups.example.com
# BrowseAllow 192.168.1.12
# BrowseAllow 192.168.1.0/24
# BrowseAllow 192.168.1.0/255.255.255.0

# BrowseDeny All
# BrowseDeny printserver.example.com
# BrowseDeny 192.168.1.13
# BrowseDeny 192.168.3.0/24
# BrowseDeny 192.168.3.0/255.255.255.0

# BrowseOrder Deny,Allow
# BrowseOrder Allow,Deny


# The interval between browsing/broadcasting cycles, local and/or
# remote, can be adjusted with the BrowseInterval directive.

# BrowseInterval 60


# Browsing-related operations such as adding or removing printer queues
# and broadcasting are each allowed to take up to a given amount of time.
# It can be configured, in seconds, with the BrowseTimeout directive.
# Especially queues discovered by CUPS broadcasts will be removed after
# this timeout if no further broadcast from the server happens.

# BrowseTimeout 300

# Filtering of remote printers by other properties than IP addresses
# of their servers

# Often the desired selection of printers cannot be reached by only
# taking into account the IP addresses of the servers. For these cases
# there is the BrowseFilter directive to filter by most of the known
# properties of the printer.

# By default there is no BrowseFilter line meaning that no filtering
# is applied.

# To do filtering one can supply one or more BrowseFilter directives
# like this:

# BrowseFilter [NOT] [EXACT] <FIELD> [<VALUE>]

# The BrowseFilter directive always starts with the word
# "BrowseFilter" and it must at least contain the name of the data
# field (<FIELD>) of the printer’s properties to which it should
# apply.

# Available field names are:

#   name:    Name of the local print queue to be created
#   host:    Host name of the remote print server
#   port:    Port through which the printer is accessed on the server
#   service: DNS/SD service name of the remote printer
#   domain:  Domain of the remote print server

# Also all field names in the TXT records of DNS-SD-advertised printers
# are valid, like "color", "duplex", "pdl", ... If the field name of
# the filter rule does not exist for the printer, the rule is skipped.

# The optional <VALUE> field is either the exact value (when the
# option EXACT is supplied) or a regular expression (Run "man 7 regex"
# in a terminal window) to be matched with the data field.

# If no <VALUE> filed is supplied, rules with field names of the TXT
# record are considered for boolean matching (true/false) of boolean
# field (like duplex, which can have the values "T" for true and "F"
# for false).

# If the option NOT is supplied, the filter rule is fulfilled if the
# regular expression or the exact value DOES NOT match the content of
# the data field. In a boolean rule (without <VALUE>) the rule matches
# false.

# Regular expressions are always considered case-insensitive and
# extended POSIX regular expressions. Field names and options (NOT,
# EXACT) are all evaluated case-insensitive. If there is an error in a
# regular expression, the BrowseFilter line gets ignored.

# Especially to note is that supplying any simple string consisting of
# only letters, numbers, spaces, and some basic special characters as
# a regular expression matches if it is contained somewhere in the
# data field.

# If there is more than one BrowseFilter directive, ALL the directives
# need to be fulfilled for the remote printer to be accepted. If one
# is not fulfilled, the printer will get ignored.

# Examples:

# Rules for standard data items which are supplied with any remote
# printer advertised via DNS-SD:

# Print queue name must contain "hum_res_", this matches
# "hum_res_mono" or "hum_res_color" but also "old_hum_res_mono":

# BrowseFilter name hum_res_

# This matches if the remote host name contains "printserver", like
# "printserver.local", "printserver2.example.com", "newprintserver":

# BrowseFilter host printserver

# This matches all ports with 631 int its number, for example 631,
# 8631, 10631,...:

# BrowseFilter port 631

# This rule matches if the DNS-SD service name contains "@ printserver":

# Browsefilter service @ printserver

# Matches all domains with "local" in their names, not only "local" but
# also things like "printlocally.com":

# BrowseFilter domain local

# Examples for rules applying to items of the TXT record:

# This rule selects PostScript printers, as the "PDL" field in the TXT
# record contains "postscript" then. This includes also remote CUPS
# queues which accept PostScript, independent of whether the physical
# printer behind the CUPS queue accepts PostScript or not.

# BrowseFilter pdl postscript

# Color printers usually contain a "Color" entry set to "T" (for true)
# in the TXT record. This rule selects them:

# BrowseFilter color

# This is a similar rule to select only duplex (automatic double-sided
# printing) printers:

# BrowseFilter duplex

# Rules with the NOT option:

# This rule EXCLUDES printers from all hosts containing "financial" in
# their names, nice to get rid of the 100s of printers of the
# financial department:

# BrowseFilter NOT host financial

# Get only monochrome printers ("Color" set to "F", meaning false, in
# the TXT record):

# BrowseFilter NOT color

# Rules with more advanced use of regular expressions:

# Only queue names which BEGIN WITH "hum_res_" are accepted now, so we
# still get "hum_res_mono" or "hum_res_color" but not
# "old_hum_res_mono" any more:

# BrowseFilter name ^hum_res_

# Server names is accepted if it contains "print_server" OR
# "graphics_dep_server":

# BrowseFilter host print_server|graphics_dep_server

# "printserver1", "printserver2", and "printserver3", nothing else:

# BrowseFilter host ^printserver[1-3]$

# Printers understanding at least one of PostScript, PCL, or PDF:

# BrowseFilter pdl postscript|pcl|pdf

# Examples for the EXACT option:

# Only printers from "printserver.local" are accepted:

# BrowseFilter EXACT host printserver.local

# Printers from all servers except "prinserver2.local" are accepted:

# BrowseFilter NOT EXACT host prinserver2.local


# Use BrowsePoll to poll a particular CUPS server

# BrowsePoll cups.example.com
# BrowsePoll cups.example.com:631
# BrowsePoll cups.example.com:631/version=1.1


# LDAP browsing configuration
# The default value for all options is an empty string. Example configuration:

# BrowseLDAPBindDN cn=cups-browsed,dc=domain,dc=tld
# BrowseLDAPCACertFile /path/to/server/certificate.pem
# BrowseLDAPDN ou=printers,dc=domain,dc=tld
# BrowseLDAPFilter (printerLocation=/Office 1/*)
# BrowseLDAPPassword s3cret
# BrowseLDAPServer ldaps://ldap.domain.tld


# Use DomainSocket to access the local CUPS daemon via another than the
# default domain socket. "None" or "Off" lets cups-browsed not use CUPS'
# domain socket.

# DomainSocket /var/run/cups/cups.sock
# DomainSocket None
# DomainSocket Off


# Set HTTP timeout (in seconds) for requests sent to local/remote
# resources Note that too short timeouts can make services getting
# missed when they are present and operations be unnecessarily
# repeated and too long timeouts can make operations take too long
# when the server does not respond.

# HttpLocalTimeout 5
# HttpRemoteTimeout 10

# Set how many retries (N) should cups-browsed do for creating print
# queues for remote printers which receive timeouts during print queue
# creation.  The printers which are not successfully set up even after
# N retries, are skipped until the next restart of the service.  Note
# that too many retries can cause high CPU load.

# HttpMaxRetries 5

# Set OnlyUnsupportedByCUPS to "Yes" will make cups-browsed not create
# local queues for remote printers for which CUPS creates queues by
# itself.  These printers are printers advertised via DNS-SD and doing
# CUPS-supported (currently PWG Raster and Apple Raster) driverless
# printing, including remote CUPS queues. Queues for other printers
# (like for legacy PostScript/PCL printers) are always created
# (depending on the other configuration settings of cups-browsed).

# With OnlyUnsupportedByCUPS set to "No", cups-browsed creates queues
# for all printers which it supports, including printers for which
# CUPS would create queues by itself. Temporary queues created by CUPS
# will get overwritten. This way it is assured that any extra
# functionality of cups-browsed will apply to these queues. As queues
# created by cups-browsed are permanent CUPS queues this setting is
# also recommended if applications/print dialogs which do not support
# temporary CUPS queues are installed. This setting is the default.

# OnlyUnsupportedByCUPS Yes


# With UseCUPSGeneratedPPDs set to "Yes" cups-browsed creates queues
# for IPP printers with PPDs generated by the PPD generator of CUPS
# and not with the one of cups-browsed. So any new development in
# CUPS' PPD generator gets available. As CUPS' PPD generator is not
# directly accessible, we need to make CUPS generate a temporary print
# queue with the desired PPD. Therefore we can only use these PPDs
# when our queue replaces a temporary CUPS queue, meaning that the
# queue is for a printer on which CUPS supports driverless printing
# (IPP 2.x, PDLs: PDF, PWG Raster, and/or Apple Raster) and that its
# name is the same as CUPS uses for the temporary queue
# ("LocalQueueNamingIPPPrinter DNS-SD" must be set). The directive
# applies only to IPP printers, not to remote CUPS queues, to not
# break clustering. Setting this directive to "No" lets cups-browsed
# generate the PPD file. Default setting is "No".
# UseCUPSGeneratedPPDs No


# With the directives LocalQueueNamingRemoteCUPS and
# LocalQueueNamingIPPPrinter you can determine how the names for local
# queues generated by cups-browsed are generated, separately for
# remote CUPS printers and IPP printers.

# DNS-SD (the default in both cases) bases the naming on the service
# name of the printer’s advertised DNS-SD record. This is exactly the
# same naming scheme as CUPS uses for its temporary queues, so the
# local queue from cups-browsed prevents CUPS from listing and
# creating an additional queue. As DNS-SD service names have to be
# unique, queue names of printers from different servers will also be
# unique and so there is no automatic clustering for load-balanced
# printing.

# MakeModel bases the queue name on the printer’s manufacturer and
# model names. This scheme cups-browsed used formerly for IPP
# printers.

# RemoteName is only available for remote CUPS queues and uses the
# name of the queue on the remote CUPS server as the local queue’s
# name. This makes printers on different CUPS servers with equal queue
# names automatically forming a load-balancing cluster as CUPS did
# formerly (CUPS 1.5.x and older) with CUPS-broadcasted remote
# printers. This scheme cups-browsed used formerly for remote CUPS
# printers.

# LocalQueueNamingRemoteCUPS DNS-SD
# LocalQueueNamingRemoteCUPS MakeModel
# LocalQueueNamingRemoteCUPS RemoteName
# LocalQueueNamingIPPPrinter DNS-SD
# LocalQueueNamingIPPPrinter MakeModel


# Set DNSSDBasedDeviceURIs to "Yes" if cups-browsed should use
# DNS-SD-service-name-based device URIs for its local queues, as CUPS
# also does. These queues use the DNS-SD service name of the
# discovered printer. With this the URI is independent of network
# interfaces and ports, giving reliable connections to always the same
# physical device. This setting is the default.

# Set DNSSDBasedDeviceURIs to "No" if cups-browsed should use the
# conventional host-name/IP-based URIs.

# Note that this option has only influence on URIs for printers
# discovered via DNS-SD, not via legacy CUPS broewsing or LDAP.
# Those printers get always assigned the conventional URIs.

# DNSSDBasedDeviceURIs Yes


# Set IPBasedDeviceURIs to "Yes" if cups-browsed should create its
# local queues with device URIs with the IP addresses instead of the
# host names of the remote servers. This mode is there for any
# problems with host name resolution in the network, especially also
# if avahi-daemon is only run for printer discovery and already
# stopped while still printing. By default this mode is turned off,
# meaning that we use URIs with host names.
# Note that the IP addresses depend on the network interface through
# which the printer is accessed. So do not use IP-based URIs on systems
# with many network interfaces and where interfaces can appear and
# disappear frequently.

# This mode could also be useful for development and debugging.

# If you prefer IPv4 or IPv6 IP addresses in the URIs, you can set
# IPBasedDeviceURIs to "IPv4" to only get IPv4 IP addresses or
# IPBasedDeviceURIs to "IPv6" to only get IPv6 IP addresses.

# IPBasedDeviceURIs No
# IPBasedDeviceURIs Yes
# IPBasedDeviceURIs IPv4
# IPBasedDeviceURIs IPv6

# The AllowResharingRemoteCUPSPrinters directive determines whether a
# print queue pointing to a remote CUPS queue will be re-shared to the
# local network or not. Since the queues generated using the BrowsePoll
# directive are also pointing to remote queues, they are also shared
# automatically if the following option is set. Default is not to share
# remote printers.

# AllowResharingRemoteCUPSPrinters Yes

# The NewBrowsePollQueuesShared directive determines whether a print
# queue for a newly discovered printer (discovered by the BrowsePoll directive)
# will be shared to the local network or not. This directive will only work
# if AllowResharingRemoteCUPSPrinters is set to yes. Default is
# not to share printers discovered using BrowsePoll.

# NewBrowsePollQueuesShared Yes

# Set CreateRemoteRawPrinterQueues to "Yes" to let cups-browsed also
# create local queues pointing to remote raw CUPS queues. Normally,
# only queues pointing to remote queues with PPD/driver are created
# as we do not use drivers on the client side, but in some cases
# accessing a remote raw queue can make sense, for example if the
# queue forwards the jobs by a special backend like Tea4CUPS.

# CreateRemoteRawPrinterQueues Yes


# cups-browsed by default creates local print queues for each shared
# CUPS print queue which it discovers on remote machines in the local
# network(s). Set CreateRemoteCUPSPrinterQueues to "No" if you do not
# want cups-browsed to do this. For example you can set cups-browsed
# to only create queues for IPP network printers setting
# CreateIPPPrinterQueues not to "No" and CreateRemoteCUPSPrinterQueues
# to "No".

# CreateRemoteCUPSPrinterQueues No


# Set CreateIPPPrinterQueues to "All" to let cups-browsed discover IPP
# network printers (native printers, not CUPS queues) with known page
# description languages (PWG Raster, PDF, PostScript, PCL XL, PCL
# 5c/e) in the local network and auto-create print queues for them.
# Set CreateIPPPrinterQueues to "Everywhere" to let cups-browsed
# discover IPP Everywhere printers in the local network (native
# printers, not CUPS queues) and auto-create print queues for them.

# Set CreateIPPPrinterQueues to "AppleRaster" to let cups-browsed
# discover Apple Raster printers in the local network (native
# printers, not CUPS queues) and auto-create print queues for them.

# Set CreateIPPPrinterQueues to "Driverless" to let cups-browsed
# discover printers designed for driverless use (currently IPP
# Everywhere and Apple Raster) in the local network (native printers,
# not CUPS queues) and auto-create print queues for them.

# Set CreateIPPPrinterQueues to "LocalOnly" to auto-create print
# queues only for local printers made available as IPP printers. These
# are for example IPP-over-USB printers, made available via
# ippusbxd. This is the default.

# Set CreateIPPPrinterQueues to "No" to not auto-create print queues
# for IPP network printers.

# The PPDs are auto-generated by cups-browsed based on properties of
# the printer polled via IPP. In case of missing information, info
# from the Bonjour record is used as last mean for the default values.

# This functionality is primarily for mobile devices running
# CUPS to not need a printer setup tool nor a collection of printer
# drivers and PPDs.

# CreateIPPPrinterQueues No
# CreateIPPPrinterQueues LocalOnly
# CreateIPPPrinterQueues Everywhere
# CreateIPPPrinterQueues AppleRaster
# CreateIPPPrinterQueues Everywhere AppleRaster
# CreateIPPPrinterQueues Driverless
# CreateIPPPrinterQueues All


# The NewIPPPrinterQueuesShared directive determines whether a print
# queue for a newly discovered IPP network printer (not remote CUPS
# queue) will be shared to the local network or not. This is only
# valid for newly discovered printers. For printers discovered in an
# earlier cups-browsed session, cups-browsed will remember whether the
# printer was shared, so changes by the user get conserved. Default is
# not to share newly discovered IPP printers.

# NewIPPPrinterQueuesShared Yes


# How to handle the print queues cups-browsed creates when
# cups-browsed is shut down:

# "KeepGeneratedQueuesOnShutdown No" makes the queues being
# removed. This makes sense as these queues only work while
# cups-browsed is running. cups-browsed has to determine to which
# member printer of a cluster to pass on the job.

# "KeepGeneratedQueuesOnShutdown Yes" (the default) makes the queues
# not being removed. This is the recommended setting for a system
# where cups-browsed is permanently running and only stopped for short
# times (like log rotation) or on shutdown. This avoids the
# re-creation of the queues when cups-browsed is restarted, which
# often causes a clutter of CUPS notifications on the desktop.

# KeepGeneratedQueuesOnShutdown No

# If there is more than one remote CUPS printer whose local queue
# would get the same name and AutoClustering is set to "Yes" (the
# default) only one local queue is created which makes up a
# load-balancing cluster of the remote printers which would get this
# queue name (implicit class). This means that when several jobs are
# sent to this queue they get distributed between the printers, using
# the method chosen by the LoadBalancing directive.

# Note that the forming of clusters depends on the naming scheme for
# local queues created by cups-browsed. If you have set
# LocalQueueNamingRemoteCUPS to "DNSSD" you will not get automatic
# clustering as the DNS-SD service names are always unique. With
# LocalQueueNamingRemoteCUPS set to "RemoteName" local queues are
# named as the CUPS queues on the remote servers are named and so
# equally named queues on different servers get clustered (this is how
# CUPS did it in version 1.5.x or older). LocalQueueNamingRemoteCUPS
# set to "MakeModel" makes remote printers of the same model get
# clustered. Note that then a cluster can contain more than one queue
# of the same server.

# With AutoClustering set to "No", for each remote CUPS printer an
# individual local queue is created, and to avoid name clashes when
# using the LocalQueueNamingRemoteCUPS settings "RemoteName" or
# "MakeModel" "@<server name>" is added to the local queue name.

# Only remote CUPS printers get clustered, not IPP network printers or
# IPP-over-USB printers.

# AutoClustering Yes
# AutoClustering No


# Load-balancing printer cluster formation can also be manually
# controlled by defining explicitly which remote CUPS printers should
# get clustered together.

# This is done by the "Cluster" directive:

# Cluster <QUEUENAME>: <EXPRESSION1> <EXPRESSION2> ...
# Cluster <QUEUENAME>

# If no expressions are given, <QUEUENAME> is used as the first and
# only expression for this cluster.

# Discovered printers are matched against all the expressions of all
# defined clusters. The first expression which matches the discovered
# printer determines to which cluster it belongs. Note that this way a
# printer can only belong to one cluster. Once matched, further
# cluster definitions will not checked any more.

# With the first printer matching a cluster’s expression a local queue
# with the name <QUEUENAME> is created. If more printers are
# discovered and match this cluster, they join the cluster. Printing
# to this queue prints to all these printers in a load-balancing
# manner, according to to the setting of the LoadBalancing directive.

# Each expression must be a string of characters without spaces. If
# spaces are needed, replace them by underscores ('_').

# An expression can be matched in three ways:

# 1. By the name of the CUPS queue on the remote server
# 2. By make and model name of the remote printer
# 3. By the DNS-SD service name of the remote printer

# Note that the matching is done case-insensitively and any group of
# non-alphanumerical characters is replaced by a single underscore.

# So if an expression is "HP_DeskJet_2540" and the remote server
# reports "hp Deskjet-2540" the printer gets matched to this cluster.

# If "AutoClustering" is not set to "No" both your manual cluster
# definitions will be followed and automatic clustering of
# equally-named remote queues will be performed. If a printer matches
# in both categories the match to the manually defined cluster has
# priority. Automatic clustering of equally-named remote printers is
# not performed if there is a manually defined cluster with this name
# (at least as the printers do not match this cluster).

# Examples:

# To cluster all remote CUPS queues named "laserprinter" in your local
# network but not cluster any other equally-named remote CUPS printers
# use (Local queue will get named "laserprinter"):

# AutoClustering No
# Cluster laserprinter

# To cluster all remote CUPS queues of HP LaserJet 4050 printers in a
# local queue named "LJ4050":

# Cluster LJ4050: HP_LaserJet_4050

# As DNS-SD service names are unique in a network you can create a
# cluster from exactly specified printers (spaces replaced by
# underscores):

# Cluster hrdep: oldlaser_@_hr-server1 newlaser_@_hr-server2


# The LoadBalancing directive switches between two methods of handling
# load balancing between equally-named remote queues which are
# represented by one local print queue making up a cluster of them
# (implicit class).

# The two methods are:

# Queuing of jobs on the client (LoadBalancing QueueOnClient):

# Here we queue up the jobs on the client and regularly check the
# clustered remote print queues. If we find an idle queue, we pass
# on a job to it.

# This is also the method which CUPS uses for classes. Advantage is a
# more even distribution of the job workload on the servers
# (especially if the printing speed of the servers is very different),
# and if a server fails, there are not several jobs stuck or
# lost. Disadvantage is that if one takes the client (laptop, mobile
# phone, ...) out of the local network, printing stops with the jobs
# waiting in the local queue.

# Queuing of jobs on the servers (LoadBalancing QueueOnServers):

# Here we check the number of jobs on each of the clustered remote
# printers and send an incoming job immediately to the remote printer
# with the lowest amount of jobs in its queue. This way no jobs queue
# up locally, all jobs which are waiting are waiting on one of the
# remote servers.

# Not having jobs waiting locally has the advantage that we can take
# the local machine from the network and all jobs get printed.
# Disadvantage is that if a server with a full queue of jobs goes
# away, the jobs go away, too.

# Default is queuing the jobs on the client as this is what CUPS does
# with classes.

# LoadBalancing QueueOnClient
# LoadBalancing QueueOnServers


# With the DefaultOptions directive one or more option settings can be
# defined to be applied to every print queue newly created by
# cups-browsed. Each option is supplied as one supplies options with
# the "-o" command line argument to the "lpadmin" command (Run "man
# lpadmin" for more details). More than one option can be supplied
# separating the options by spaces. By default no option settings are
# pre-defined.

# Note that print queues which cups-browsed already created before
# remember their previous settings and so these settings do not get
# applied.

# DefaultOptions Option1=Value1 Option2=Value2 Option3 noOption4


# The AutoShutdown directive specifies whether cups-browsed should
# automatically terminate when it has no local raw queues set up
# pointing to any discovered remote printers or no jobs on such queues
# depending on AutoShutdownOn setting (auto shutdown mode). Setting it
# to "On" activates the auto-shutdown mode, setting it to "Off"
# deactiivates it (the default). The special mode "avahi" turns auto
# shutdown off while avahi-daemon is running and on when avahi-daemon
# stops. This allows running cups-browsed on-demand when avahi-daemon
# is run on-demand.

# AutoShutdown Off
# AutoShutdown On
# AutoShutdown avahi


# The AutoShutdownOn directive determines what event cups-browsed
# considers as inactivity in auto shutdown mode. "NoQueues" (the
# default) means that auto shutdown is initiated when there are no
# queues for discovered remote printers generated by cups-browsed any
# more. "NoJobs" means that all queues generated by cups-browsed are
# without jobs.

# AutoShutdownOn NoQueues
# AutoShutdownOn NoJobs


# The AutoShutdownTimeout directive specifies after how many seconds
# without local raw queues set up pointing to any discovered remote
# printers or jobs on these queues cups-browsed should actually shut
# down in auto shutdown mode. Default is 30 seconds, 0 means immediate
# shutdown.

# AutoShutdownTimeout 30

# DebugLogFileSize defines the maximum size possible (in KBytes)
# of the log files (cups-browsed_log and cups-browsed_previous_logs)
# that is created using cups-browsed in the debugging mode.
# Setting its value to 0 would turn off any restriction
# on the size of the file.

# DebugLogFileSize 300

# NotifLeaseDuration defines how long the D-BUS subscription created by cups-browsed
# in cupsd will last before cupsd cancels it. The default value is 1 day
# in seconds - 86400. The subscription renewal is set to happen after half of
# NotifLeaseDuration passed. The D-BUS notifications are used for watching over queues
# and doing specific actions when a D-BUS notification comes.

# NotifLeaseDuration 86400

# FrequentNetifUpdate turns on/off the network interface update routines
# which happen for each found entry, which can slow up cups-browsed significantly
# if we are on a network with many shared printers or if we use BrowsePoll to a server
# with many queues. Network interface updates after receiving D-BUS notification
# from NetworkManager won’t be turned off with the directive. The default value
# is ’Yes’.
#
# FrequentNetifUpdate Yes
```