NAME

auto.master – Master Map for automounter

DESCRIPTION

The **auto.master** map is consulted to set up automount managed mount points when the **autofs**(8) script is invoked or the **automount**(8) program is run. Each line describes a mount point and refers to an autofs map describing file systems to be mounted under the mount point.

The default location of the master map is **/etc/auto.master** but an alternate name may be given on the command line when running the automounter and the default master map may changed by setting the **MASTER_MAP_NAME** configuration variable in **/etc/sysconfig/autofs.** If the master map name has no path then the system Name Service Switch configuration will be consulted and each of the sources searched in line with the rules given in the Name Service Switch configuration.

Access to mounts in maps is governed by a key.

For direct maps the mount point is always specified as:

/-

and the key used within the direct map is the full path to the mount point. The direct map may have multiple entries in the master map.

For indirect maps access is by using the path scheme:

/mount-point/key

where *mount-point* is one of the entries listed in the master map. The *key* is a single directory component and is matched against entries in the map given in the entry (See **autofs**(5)).

Additionally, a map may be included from its source as if it were itself present in the master map by including a line of the form:

+[maptype[,**format**]:]**map** [**options**]

and **automount**(8) will process the map according to the specification described below for map entries. Indirect map entries must be unique in the master map so second and subsequent entries for an indirect mount point are ignored by **automount**(8).

FORMAT

Master map entries have three fields separated by an arbitrary number of spaces or tabs. Lines beginning with # are comments. The first field is the mount point described above and the second field is the name of the map to be consulted for the mount point followed by the third field which contains options to be applied to all entries in the map.

The format of a master map entry is:

mount-point [**map-type**[,**format**]:]**map** [**options**]

mount-point

Base location for the **autofs** filesystem to be mounted. For indirect maps this directory will be created (as with **mkdir −p**) and is removed when the **autofs** filesystem is umounted.

map-type

Type of map used for this mount point. The following are valid map types:

file The map is a regular text file.

program

The map is an executable program, which is passed a key on the command line and returns an entry (everything besides the key) on stdout if successful. Optinally, the keyword exec may be used as a synonym for program to avoid confusion with amd formated maps mount type program.

yp The map is a NIS (YP) database.

nisplus The map is a NIS+ database.

hesiod The map is a hesiod database whose **filsys** entries are used for maps.

ldap or **ldaps**

> The map is stored in an LDAP directory. If **ldaps** is used the appropriate certificate must be configured in the LDAP client.

multi This map type allows the specification of multiple maps separated by "--". These maps are searched in order to resolve key lookups.

format

Format of the map data; currently the formats recognized are **sun**, which is a subset of the Sun automounter map format, **hesiod**, for hesiod filesys entries and **amd** for amd formated map entries. If the format is left unspecified, it defaults to **sun** for all map types except **hesiod** unless it is a top level **amd** mount that has a configuration entry for the mount point path, in which case the format used is **amd**.

map

Name of the map to use. This is an absolute UNIX pathname for maps of types **file** or **program**, and the name of a database in the case for maps of type **yp**, **nisplus**, or **hesiod** or the **dn** of an LDAP entry for maps of type **ldap**.

options

Any remaining command line arguments without leading dashes (−) are taken as options (−*o*) to **mount**. Arguments with leading dashes are considered options for the maps and are passed to automount (8).

The **sun** format supports the following options:

−*Dvariable=value*

> Replace *variable* with *value* in map substitutions.

−*strict* Treat errors when mounting file systems as fatal. This is important when multiple file systems should be mounted ('multimounts'). If this option is given, no file system is mounted at all if at least one file system can't be mounted.

[no]browse

> This is an autofs specific option that is a pseudo mount option and so is given without a leading dash. Use of the browse option pre-creates mount point directories for indirect mount maps so the map keys can be seen in a directory listing without being mounted. Use of this option can cause performance problem if the indirect map is large so it should be used with caution. The internal program default is to enable browse mode for indirect mounts but the default installed configuration overrides this by setting BROWSE_MODE to "no" because of the potential performance problem.

nobind This is an autofs specific option that is a pseudo mount option and so is given without a leading dash. It may be used either in the master map entry (so it effects all the map entries) or with individual map entries to prevent bind mounting of local NFS filesystems. For direct mount maps the option is only effective if specified on the first direct map entry and is applied to all direct mount maps in the master map. It is ignored if given on subsequent direct map entries. It may be used on individual map entries of both types. Bind mounting of NFS file systems can also be prevented for specific map entrys by adding the "port=" mount option to the entries.

symlink

> This option makes bind mounting use a symlink instead of an actual bind mount. It is an autofs specific option that is a pseudo mount option and so is given without a leading dash. It may be used with indirect map entries only, either in the master map (so it effects all map entries) or with individual map entries. The option is ignored for direct mounts and non-root offest mount entries.

> −r, −−random-multimount-selection
>> Enables the use of ramdom selection when choosing a host from a list of replicated servers. This option is applied to this mount only, overriding the global setting that may be specified on the command line.
>
> −w, −−use-weight-only
>> Use only specified weights for server selection where more than one server is specified in the map entry. If no server weights are given then each available server will be tried in the order listed, within proximity.
>
> −t, −−timeout <seconds>
>> Set the expire timeout for map entries. This option can be used to override the global default given either on the command line or in the configuration.
>
> −n, −−negative−timeout <seconds>
>> Set the timeout for caching failed key lookups. This option can be used to override the global default given either on the command line or in the configuration.

BUILTIN MAP -hosts

If "-hosts" is given as the map then accessing a key under the mount point which corresponds to a hostname will allow access to the exports of that host. The hosts map cannot be dynamically updated and requires a HUP signal to be sent to the daemon for it to check hosts for an update. Due to possible hierarchic dependencies within a mount tree, it might not be completely updated during the HUP signal processing.

For example, with an entry in the master map of **/net -hosts** accessing /net/myserver will mount exports from myserver on directories below /net/myserver.

NOTE: mounts done from a hosts map will be mounted with the "nosuid,nodev,intr" options unless overridden by explicitly specifying the "suid", "dev" or "nointr" options in the master map entry.

LDAP MAPS

If the map type **ldap** is specified the mapname is of the form **[//servername/]dn**, where the optional **servername** is the name of the LDAP server to query, and **dn** is the Distinguished Name of a subtree to search for map entries. The old style **ldap:servername:mapname** is also understood. Alternatively, the type can be obtained from the Name Service Switch configuration, in which case the map name alone must be given.

If no schema is set in the autofs configuration then autofs will check each of the commonly used schema for a valid entry and if one is found it will used for subsequent lookups.

There are three common schemas in use:

nisMap
> Entries in the **nisMap** schema are **nisObject** objects in the specified subtree, where the **cn** attribute is the key (the wildcard key is "/"), and the **nisMapEntry** attribute contains the information used by the automounter.

automountMap
> The **automountMap** schema has two variations that differ in the attribute used for the map key. Entries in the automountMap schema are **automount** objects in the specified subtree, where the **cn** or **automountKey** attribute (depending on local usage) is the key (the wildcard key is "/"), and the **automountInformation** attribute contains the information used by the automounter. Note that the **cn** attribute is case insensitive.

The object classes and attributes used for accessing automount maps in LDAP can be changed by setting entries in the autofs configuration located in **/etc/sysconfig/autofs.conf**.

NOTE:
> If a schema is given in the configuration then all the schema configuration values must be set, any partial schema specification will be ignored.

For **amd** format maps a different schema is used:

amdMap

> The **amdmap** schema contains attributes **amdmapName**, **amdmapKey** and **amdmapValue** where **amdmapName** contains the name of the containing map, **amdmapKey** contains the map key and **amdmapValue** contains the map entry.

LDAP AUTHENTICATION, ENCRYPTED AND CERTIFIED CONNECTIONS

LDAP authenticated binds, TLS encrypted connections and certification may be used by setting appropriate values in the autofs authentication configuration file and configuring the LDAP client with appropriate settings. The default location of this file is **/etc/autofs_ldap_auth.conf**.

If this file exists it will be used to establish whether TLS or authentication should be used.

An example of this file is:

```
<?xml version="1.0" ?>
<autofs_ldap_sasl_conf
    usetls="yes"
    tlsrequired="no"
    authrequired="no"
    authtype="DIGEST-MD5"
    user="xyz"
    secret="abc"
/>
```

If TLS encryption is to be used the location of the Certificate Authority certificate must be set within the LDAP client configuration in order to validate the server certificate. If, in addition, a certified connection is to be used then the client certificate and private key file locations must also be configured within the LDAP client.

In OpenLDAP these may be configured in the **ldap.conf** file or in the per-user configuration. For example it may be sensible to use the system wide configuration for the location of the Certificate Authority certificate and set the location of the client certificate and private key in the per-user configuration. The location of these files and the configuration entry requirements is system dependent so the documentation for your installation will need to be consulted to get further information.

See **autofs_ldap_auth.conf (5)** for more information.

EXAMPLE

```
/-              auto.data
/home           /etc/auto.home
/mnt            yp:mnt.map
```

This will generate two mountpoints for */home* and */mnt* and install direct mount triggers for each entry in the direct mount map *auto.data*. All accesses to */home* will lead to the consultation of the map in */etc/auto.home* and all accesses to */mnt* will consult the NIS map *mnt.map*. All accesses to paths in the map *auto.data* will trigger mounts when they are accessed and the Name Service Switch configuration will be used to locate the source of the map *auto.data*.

SEE ALSO

automount(8), **autofs**(5), **autofs**(8), **autofs.conf**(5), **autofs_ldap_auth.conf**(5)

AUTHOR

This manual page was written by Christoph Lameter <chris@waterf.org>, for the Dean GNU/Linux system. Edited by <hpa@transmeta.com> and Ian Kent <raven@themaw.net> .