NAME

nsswitch.conf – Name Service Switch configuration file

DESCRIPTION

The Name Service Switch (NSS) configuration file, */etc/nsswitch.conf*, is used by the GNU C Library and certain other applications to determine the sources from which to obtain name-service information in a range of categories, and in what order. Each category of information is identified by a database name.

The file is plain ASCII text, with columns separated by spaces or tab characters. The first column specifies the database name. The remaining columns describe the order of sources to query and a limited set of actions that can be performed by lookup result.

The following databases are understood by the GNU C Library:

aliases Mail aliases, used by **getaliasent**(3) and related functions.

ethers Ethernet numbers.

group Groups of users, used by **getgrent**(3) and related functions.

hosts Host names and numbers, used by **gethostbyname**(3) and related functions.

initgroups Supplementary group access list, used by **getgrouplist**(3) function.

netgroup Network-wide list of hosts and users, used for access rules. C libraries before glibc 2.1 supported netgroups only over NIS.

networks Network names and numbers, used by **getnetent**(3) and related functions.

passwd User passwords, used by **getpwent**(3) and related functions.

protocols Network protocols, used by **getprotoent**(3) and related functions.

publickey Public and secret keys for Secure_RPC used by NFS and NIS+.

rpc Remote procedure call names and numbers, used by **getrpcbyname**(3) and related functions.

services Network services, used by **getservent**(3) and related functions.

shadow Shadow user passwords, used by **getspnam**(3) and related functions.

The GNU C Library ignores databases with unknown names. Some applications use this to implement special handling for their own databases. For example, **sudo**(8) consults the **sudoers** database.

Here is an example */etc/nsswitch.conf* file:

```
passwd:         compat
group:          compat
shadow:         compat

hosts:          dns [!UNAVAIL=return] files
networks:       nis [NOTFOUND=return] files
ethers:         nis [NOTFOUND=return] files
protocols:      nis [NOTFOUND=return] files
rpc:            nis [NOTFOUND=return] files
services:       nis [NOTFOUND=return] files
```

The first column is the database name. The remaining columns specify:

* One or more service specifications, for example, "files", "db", or "nis". The order of the services on the line determines the order in which those services will be queried, in turn, until a result is found.
* Optional actions to perform if a particular result is obtained from the preceding service, for example, "[NOTFOUND=return]".

The service specifications supported on your system depend on the presence of shared libraries, and are therefore extensible. Libraries called */lib/libnss_SERVICE.so.***X** will provide the named *SERVICE*. On a

standard installation, you can use "files", "db", "nis", and "nisplus". For the **hosts** database, you can additionally specify "dns". For the **passwd**, **group**, and **shadow** databases, you can additionally specify "compat" (see **Compatibility mode** below). The version number **X** may be 1 for glibc 2.0, or 2 for glibc 2.1 and later. On systems with additional libraries installed, you may have access to further services such as "hesiod", "ldap", "winbind" and "wins".

An action may also be specified following a service specification. The action modifies the behavior following a result obtained from the preceding data source. Action items take the general form:

[*STATUS*=*ACTION*]
[!*STATUS*=*ACTION*]

where

STATUS => **success** | **notfound** | **unavail** | **tryagain**
ACTION => **return** | **continue** | **merge**

The ! negates the test, matching all possible results except the one specified. The case of the keywords is not significant.

The *STATUS* value is matched against the result of the lookup function called by the preceding service specification, and can be one of:

success No error occurred and the requested entry is returned. The default action for this condition is "return".

notfound The lookup succeeded, but the requested entry was not found. The default action for this condition is "continue".

unavail The service is permanently unavailable. This can mean either that the required file cannot be read, or, for network services, that the server is not available or does not allow queries. The default action for this condition is "continue".

tryagain The service is temporarily unavailable. This could mean a file is locked or a server currently cannot accept more connections. The default action for this condition is "continue".

The *ACTION* value can be one of:

return Return a result now. Do not call any further lookup functions. However, for compatibility reasons, if this is the selected action for the **group** database and the **notfound** status, and the configuration file does not contain the **initgroups** line, the next lookup function is always called, without affecting the search result.

continue Call the next lookup function.

merge *[SUCCESS=merge]* is used between two database entries. When a group is located in the first of the two group entries, processing will continue on to the next one. If the group is also found in the next entry (and the group name and GID are an exact match), the member list of the second entry will be added to the group object to be returned. Available since glibc 2.24. Note that merging will not be done for **getgrent**(3) nor will duplicate members be pruned when they occur in both entries being merged.

Compatibility mode (compat)

The NSS "compat" service is similar to "files" except that it additionally permits special entries in corresponding files for granting users or members of netgroups access to the system. The following entries are valid in this mode:

For **passwd** and **shadow** databases:

+*user* Include the specified *user* from the NIS passwd/shadow map.

+*user*:::::: Include the specified *user* from the NIS passwd map, but override with non-empty passwd fields.

+@*netgroup* Include all users in the given *netgroup*.

−*user* Exclude the specified *user* from the NIS passwd/shadow map.

−@*netgroup* Exclude all users in the given *netgroup*.

+ Include every user, except previously excluded ones, from the NIS passwd/shadow map.

For **group** database:

+*group* Include the specified *group* from the NIS group map.

−*group* Exclude the specified *group* from the NIS group map.

+ Include every group, except previously excluded ones, from the NIS group map.

By default, the source is "nis", but this may be overridden by specifying any NSS service except "compat" itself as the source for the pseudo-databases **passwd_compat**, **group_compat**, and **shadow_compat**.

FILES

A service named *SERVICE* is implemented by a shared object library named *libnss_SERVICE.so.***X** that resides in */lib*.

/etc/nsswitch.conf	NSS configuration file.
*/lib/libnss_compat.so.***X**	implements "compat" source.
*/lib/libnss_db.so.***X**	implements "db" source.
*/lib/libnss_dns.so.***X**	implements "dns" source.
*/lib/libnss_files.so.***X**	implements "files" source.
*/lib/libnss_hesiod.so.***X**	implements "hesiod" source.
*/lib/libnss_nis.so.***X**	implements "nis" source.
*/lib/libnss_nisplus.so.***X**	implements "nisplus" source.

The following files are read when "files" source is specified for respective databases:

aliases	*/etc/aliases*
ethers	*/etc/ethers*
group	*/etc/group*
hosts	*/etc/hosts*
initgroups	*/etc/group*
netgroup	*/etc/netgroup*
networks	*/etc/networks*
passwd	*/etc/passwd*
protocols	*/etc/protocols*
publickey	*/etc/publickey*
rpc	*/etc/rpc*
services	*/etc/services*
shadow	*/etc/shadow*

NOTES

Within each process that uses **nsswitch.conf**, the entire file is read only once. If the file is later changed, the process will continue using the old configuration.

Traditionally, there was only a single source for service information, often in the form of a single configuration file (e.g., */etc/passwd*). However, as other name services, such as the Network Information Service (NIS) and the Domain Name Service (DNS), became popular, a method was needed that would be more flexible than fixed search orders coded into the C library. The Name Service Switch mechanism, which was based on the mechanism used by Sun Microsystems in the Solaris 2 C library, introduced a cleaner solution to the problem.

SEE ALSO

getent(1), **nss**(5)

COLOPHON

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.