NAME

netconfig – network configuration database

SYNOPSIS

/etc/netconfig

DESCRIPTION

The network configuration database, **/etc/netconfig**, is a system file used to store information about networks that are connected to the system. The **netconfig** database and the routines that access it (see **getnetconfig**(3N)) are part of the Network Selection component. The Network Selection component also includes **getnetpath**(3N) routines to provide application-specific network search paths. These routines access the **netconfig** database based on the environment variable **NETPATH** (see **environ**(5)).

netconfig contains an entry for each network available on the system. Entries are separated by newlines. Fields are separated by whitespace and occur in the order in which they are described below. Whitespace can be embedded as ''*blank*'' or ''*tab*''. Backslashes may be embedded as ''\\\\''. Lines in **/etc/netconfig** that begin with a **#** (hash) in column 1 are treated as comments.

Each of the valid lines in the **netconfig** database correspond to an available transport. Each entry is of the form:

network ID semantics flag protocol-family protocol-name network-device translation-libraries

network ID
: A string used to uniquely identify a network. *network ID* consists of non-null characters, and has a length of at least 1. No maximum length is specified. This namespace is locally significant and the local system administrator is the naming authority. All *network ID*s on a system must be unique.

semantics
: The *semantics* field is a string identifying the ''semantics'' of the network, that is, the set of services it supports, by identifying the service interface it provides. The *semantics* field is mandatory. The following semantics are recognized.

  tpi_clts Transport Provider Interface, connectionless

  tpi_cots Transport Provider Interface, connection oriented

  tpi_cots_ord Transport Provider Interface, connection oriented, supports orderly release.

flag
: The *flag* field records certain two-valued (''true'' and ''false'') attributes of networks. *flag* is a string composed of a combination of characters, each of which indicates the value of the corresponding attribute. If the character is present, the attribute is ''true.'' If the character is absent, the attribute is ''false.'' ''**-**'' indicates that none of the attributes are present. Only one character is currently recognized:

  v Visible (''default'') network. Used when the environment variable **NETPATH** is unset.

protocol family

The *protocol family* and *protocol name* fields are provided for protocol-specific applications.

The *protocol family* field contains a string that identifies a protocol family. The *protocol family* identifier follows the same rules as those for *network ID*s; the string consists of non-null characters, it has a length of at least **1**, and there is no maximum length specified. A ''–'' in the *protocol family* field indicates that no protocol family identifier applies (the network is experimental). The following are examples:

loopback	Loopback (local to host).
inet	Internetwork: UDP, TCP, etc.
implink	ARPANET imp addresses
pup	PUP protocols: for example, BSP
chaos	MIT CHAOS protocols
ns	XEROX NS protocols
nbs	NBS protocols
ecma	European Computer Manufacturers Association
datakit	DATAKIT protocols
ccitt	CCITT protocols, X.25, etc.
sna	IBM SNA
decnet	DECNET
dli	Direct data link interface
lat	LAT
hylink	NSC Hyperchannel
appletalk	Apple Talk
nit	Network Interface Tap
ieee802	IEEE 802.2; also ISO 8802
osi	Umbrella for all families used by OSI (for example, protosw lookup)
x25	CCITT X.25 in particular
osinet	AFI = 47, IDI = 4
gosip	U.S. Government OSI

protocol name The *protocol name* field contains a string that identifies a protocol. The *protocol name* identifier follows the same rules as those for *network ID*s; that is, the string consists of non-NULL characters, it has a length of at least **1**, and there is no maximum length specified. A ''–'' indicates that none of the names listed apply. The following protocol names are recognized.

tcp	Transmission Control Protocol
udp	User Datagram Protocol
icmp	Internet Control Message Protocol

network device
The *network device* is the full pathname of the device used to connect to the transport provider. Typically, this device will be in the **/dev** directory. The *network device* must be specified.

translation libraries
The *name-to-address translation libraries* support a ''directory service'' (a name-to-address mapping service) for the network. A ''–'' in this field indicates the absence of any *translation libraries*. This has a special meaning for networks of the protocol family **inet :** its name-to-address mapping is provided by the name service switch based on the entries for **hosts** and **services** in **nsswitch.conf**(4). For networks of other families, a ''–'' indicates non-functional name-to-address mapping. Otherwise, this field consists of a comma-separated list of pathnames to dynamically linked libraries. The pathname of the library can be either absolute or relative. See **dlopen**(3X).

Each field corresponds to an element in the **struct netconfig** structure. **struct netconfig** and the identifiers described on this manual page are defined in **<netconfig.h>**. This structure includes the following members:

char **nc_netid*	Network ID, including NULL terminator.
unsigned long *nc_semantics*	Semantics.
unsigned long *nc_flag*	Flags.
char **nc_protofmly*	Protocol family.
char **nc_proto*	Protocol name.
char **nc_device*	Full pathname of the network device.
unsigned long *nc_nlookups*	Number of directory lookup libraries.
char ***nc_lookups*	Names of the name-to-address translation libraries.
unsigned long *nc_unused[9]*	Reserved for future expansion.

The *nc_semantics* field takes the following values, corresponding to the semantics identified above:

NC_TPI_CLTS
NC_TPI_COTS
NC_TPI_COTS_ORD

The *nc_flag* field is a bitfield. The following bit, corresponding to the attribute identified above, is currently recognized. **NC_NOFLAG** indicates the absence of any attributes.

NC_VISIBLE

EXAMPLES

Below is a sample **netconfig** file:

```
#
# The "Network Configuration" File.
#
# Each entry is of the form:
#
#   <network_id> <semantics> <flags> <protofamily> <protoname> <device> \
#        <nametoaddr_libs>
#
# The "-" in <nametoaddr_libs> for inet family transports indicates
# redirection to the name service switch policies for "hosts" and
# "services". The "-" may be replaced by nametoaddr libraries that
# comply with the SVr4 specs, in which case the name service switch
# will not be used for netdir_getbyname, netdir_getbyaddr,
# gethostbyname, gethostbyaddr, getservbyname, and getservbyport.
# There are no nametoaddr_libs for the inet family in Solaris anymore.
#
udp        tpi_clts      v  inet      udp  /dev/udp       -
tcp        tpi_cots_ord  v  inet      tcp  /dev/tcp       -
rawip      tpi_raw       -  inet      -    /dev/rawip     -
ticlts     tpi_clts      v  loopback  -    /dev/ticlts    straddr.so
ticotsord  tpi_cots_ord  v  loopback  -    /dev/ticotsord straddr.so
ticots     tpi_cots      v  loopback  -    /dev/ticots    straddr.so
```

FILES

<netconfig.h>

SEE ALSO

dlopen(3X), **getnetconfig**(3N), **getnetpath**(3N), **nsswitch.conf**(4)

NFS Administration Guide
Transport Interfaces Programming Guide