

**NAME**

**bootptab** - Internet Bootstrap Protocol server database

**DESCRIPTION**

The **bootptab** file is the configuration database file for bootpd(8), the Internet Bootstrap Protocol server. Its format is similar to that of termcap(5) in which two-character case-sensitive tag symbols are used to represent host parameters. These parameter declarations are separated by colons (:), with a general format of:

```
hostname:tg=value. . . :tg=value. . . :tg=value. . . .
```

where *hostname* is the actual name of a bootp client (or a "dummy entry"), and *tg* is a two-character tag symbol. Dummy entries have an invalid hostname (one with a "." as the first character) and are used to provide default values used by other entries via the *tc=.dummy-entry* mechanism. Most tags must be followed by an equals-sign and a value as above. Some may also appear in a boolean form with no value (i.e. *:tg:*). The currently recognized tags are:

bf	Bootfile
bs	Bootfile size in 512-octet blocks
cs	Cookie server address list
df	Merit dump file
dn	Domain name
ds	Domain name server address list
ef	Extension file
gw	Gateway address list
ha	Host hardware address
hd	Bootfile home directory
hn	Send client's hostname to client
ht	Host hardware type (see Assigned Numbers RFC)
im	Impress server address list
ip	Host IP address
lg	Log server address list
lp	LPR server address list
ns	IEN-116 name server address list
nt	NTP (time) Server (RFC 1129)
ra	Reply address override
rl	Resource location protocol server address list
rp	Root path to mount as root
sa	TFTP server address client should use
sm	Host subnet mask

sw Swap server address  
tc Table continuation (points to similar "template" host entry)  
td TFTP root directory used by "secure" TFTP servers  
to Time offset in seconds from UTC  
ts Time server address list  
vm Vendor magic cookie selector  
yd YP (NIS) domain name  
ys YP (NIS) server address

There is also a generic tag, *Tn*, where *n* is an RFC1084 vendor field tag number. Thus it is possible to immediately take advantage of future extensions to RFC1084 without being forced to modify **bootpd** first. Generic data may be represented as either a stream of hexadecimal numbers or as a quoted string of ASCII characters. The length of the generic data is automatically determined and inserted into the proper field(s) of the RFC1084-style bootp reply.

The following tags take a whitespace-separated list of IP addresses: *cs*, *ds*, *gw*, *im*, *lg*, *lp*, *ns*, *nt*, *ra*, *rl*, and *ts*. The *ip*, *sa*, *sw*, *sm*, and *ys* tags each take a single IP address. All IP addresses are specified in standard Internet "dot" notation and may use decimal, octal, or hexadecimal numbers (octal numbers begin with 0, hexadecimal numbers begin with '0x' or '0X'). Any IP addresses may alternatively be specified as a hostname, causing **bootpd** to lookup the IP address for that host name using `gethostbyname(3)`. If the *ip* tag is not specified, **bootpd** will determine the IP address using the entry name as the host name. (Dummy entries use an invalid host name to avoid automatic IP lookup.)

The *ht* tag specifies the hardware type code as either an unsigned decimal, octal, or hexadecimal integer or one of the following symbolic names: *ethernet* or *ether* for 10Mb Ethernet, *ethernet3* or *ether3* for 3Mb experimental Ethernet, *ieee802*, *tr*, or *token-ring* for IEEE 802 networks, *pronet* for Proteon ProNET Token Ring, or *chaos*, *arcnet*, or *ax.25* for Chaos, ARCNET, and AX.25 Amateur Radio networks, respectively. The *ha* tag takes a hardware address which may be specified as a host name or in numeric form. Note that the numeric form *must* be specified in hexadecimal; optional periods and/or a leading '0x' may be included for readability. The *ha* tag must be preceded by the *ht* tag (either explicitly or implicitly; see *tc* below). If the hardware address is not specified and the type is specified as either "ethernet" or "ieee802", then **bootpd** will try to determine the hardware address using `ether_hostton(3)`.

The hostname, home directory, and bootfile are ASCII strings which may be optionally surrounded by double quotes ("). The client's request and the values of the *hd* and *bf* symbols determine how the server fills in the bootfile field of the bootp reply packet.

If the client provides a file name it is left as is. Otherwise, if the *bf* option is specified its value is copied into the reply packet. If the *hd* option is specified as well, its value is prepended to the boot file copied

into the reply packet. The existence of the boot file is checked only if the *bs=auto* option is used (to determine the boot file size). A reply may be sent whether or not the boot file exists.

Some newer versions of `tftpd(8)` provide a security feature to change their root directory using the `chroot(2)` system call. The *td* tag may be used to inform **bootpd** of this special root directory used by **tftpd**. (One may alternatively use the **bootpd -c *chdir*** option.) The *hd* tag is actually relative to the root directory specified by the *td* tag. For example, if the real absolute path to your BOOTP client bootfile is */tftpboot/bootfiles/bootimage*, and **tftpd** uses */tftpboot* as its "secure" directory, then specify the following in *bootptab*:

```
:td=/tftpboot:hd=/bootfiles:bf=bootimage:
```

If your bootfiles are located directly in */tftpboot*, use:

```
:td=/tftpboot:hd=:bf=bootimage:
```

The *sa* tag may be used to specify the IP address of the particular TFTP server you wish the client to use. In the absence of this tag, **bootpd** will tell the client to perform TFTP to the same machine **bootpd** is running on.

The time offset *to* may be either a signed decimal integer specifying the client's time zone offset in seconds from UTC, or the keyword *auto* which uses the server's time zone offset. Specifying the *to* symbol as a boolean has the same effect as specifying *auto* as its value.

The bootfile size *bs* may be either a decimal, octal, or hexadecimal integer specifying the size of the bootfile in 512-octet blocks, or the keyword *auto* which causes the server to automatically calculate the bootfile size at each request. As with the time offset, specifying the *bs* symbol as a boolean has the same effect as specifying *auto* as its value.

The vendor magic cookie selector (the *vm* tag) may take one of the following keywords: *auto* (indicating that vendor information is determined by the client's request), *rfc1048* or *rfc1084* (which always forces an RFC1084-style reply), or *cmu* (which always forces a CMU-style reply).

The *hn* tag is strictly a boolean tag; it does not take the usual equals-sign and value. Its presence indicates that the hostname should be sent to RFC1084 clients. **Bootpd** attempts to send the entire hostname as it is specified in the configuration file; if this will not fit into the reply packet, the name is shortened to just the host field (up to the first period, if present) and then tried. In no case is an arbitrarily-truncated hostname sent (if nothing reasonable will fit, nothing is sent).

Often, many host entries share common values for certain tags (such as name servers, etc.). Rather than

repeatedly specifying these tags, a full specification can be listed for one host entry and shared by others via the *tc* (table continuation) mechanism. Often, the template entry is a dummy host which does not actually exist and never sends bootp requests. This feature is similar to the *tc* feature of *termcap(5)* for similar terminals. Note that **bootpd** allows the *tc* tag symbol to appear anywhere in the host entry, unlike *termcap* which requires it to be the last tag. Information explicitly specified for a host always overrides information implied by a *tc* tag symbol, regardless of its location within the entry. The value of the *tc* tag may be the hostname or IP address of any host entry previously listed in the configuration file.

Sometimes it is necessary to delete a specific tag after it has been inferred via *tc*. This can be done using the construction *tag@* which removes the effect of *tag* as in *termcap(5)*. For example, to completely undo an IEN-116 name server specification, use *:ns@:* at an appropriate place in the configuration entry. After removal with *@*, a tag is eligible to be set again through the *tc* mechanism.

Blank lines and lines beginning with "#" are ignored in the configuration file. Host entries are separated from one another by newlines; a single host entry may be extended over multiple lines if the lines end with a backslash (\). It is also acceptable for lines to be longer than 80 characters. Tags may appear in any order, with the following exceptions: the hostname must be the very first field in an entry, and the hardware type must precede the hardware address.

An example */etc/bootptab* file follows:

```
# Sample bootptab file (domain=andrew.cmu.edu)
```

```
.default:\
    :hd=/usr/boot:bf=null:\
    :ds=netserver, lancaster:\
    :ns=pcs2, pcs1:\
    :ts=pcs2, pcs1:\
    :sm=255.255.255.0:\
    :gw=gw.cs.cmu.edu:\
    :hn:to=-18000:

carnegie:ht=6:ha=7FF810000AF:tc=.default:
baldwin:ht=1:ha=0800200159C3:tc=.default:
wylie:ht=1:ha=00DD00CADF00:tc=.default:
arnold:ht=1:ha=0800200102AD:tc=.default:
bairdford:ht=1:ha=08002B02A2F9:tc=.default:
bakerstown:ht=1:ha=08002B0287C8:tc=.default:
```

```
# Special domain name server and option tags for next host
```

```
butlerjct:ha=08002001560D:ds=128.2.13.42:\
:T37=0x12345927AD3BCF:\
:T99="Special ASCII string":\
:tc=.default:
```

```
gastonville:ht=6:ha=7FFF81000A47:tc=.default:
hahntown:ht=6:ha=7FFF81000434:tc=.default:
hickman:ht=6:ha=7FFF810001BA:tc=.default:
lowber:ht=1:ha=00DD00CAF000:tc=.default:
mtoliver:ht=1:ha=00DD00FE1600:tc=.default:
```

**FILES**

*/etc/bootptab*

**SEE ALSO**

bootpd(8), tftpd(8)

DARPA Internet Request For Comments RFC951, RFC1048, RFC1084, Assigned Numbers