

NEXTSTEP

Title: bootpd Information Sources

Entry Number: 1993

Creation Date: September 6 1995

Last Updated: <<Date September 6 1995>>

Keywords: netboot, bootptab, bootpd, kernel, tftpboot, directory

Question

From where does a netboot server (bootpd) get its "served kernel" information?

Answer

The netboot server gets its kernel information from two sources:

- NetInfo and
- /etc/bootptab.

(Yes, from both. Even if NetInfo is running, the file is consulted.)

There are two types of bootpd information: global (i.e., for all netboot clients) and local (i.e., for a specific client). The global information is read from the file /etc/bootptab; the local information comes from NetInfo (the /machines/client directory).

There are two pieces of global information: the tftpboot root directory, and the default kernel to be served. [What's tftpboot? It's the file transfer mechanism used for netbooting Ð trivial file transfer protocol boot.] All kernels which can be booted must be located in the tftpboot root directory, which is typically /private/tftpboot on our system. Any pathname given, either for the default kernel or for a per-client kernel, is relative to the tftpboot root directory; further, that directory is considered as the root directory for the tftpboot process, so a path like /sdmach will be interpreted

as `/private/tftpboot/sdmach`. (Why is this done? Security: it prevents booting arbitrary files from the boot server.)

Then there's the local information: a host type, ethernet address, IP address, and bootfile. These are all obtained from `NetInfo`. Note that, although the host type appears in `bootptab`, it is not present in `NetInfo`; if an `niload(8)` is done, it MUST have the value 1 (else `niload` will yield an error). If you `nidump bootptab`, a 1 will be provided in the field.

Now, the stage is set for explaining how the netboot sequence occurs, regarding the boot protocol.

The following paragraph describes the boot sequence from the server's perspective:

1. A client sends a BOOTP broadcast, asking for the file `boot` (the name is in the PROM). This is the request for the `blk0` loader, also known as the `bootstrap loader`. The server notices same.
2. The server then checks in `NetInfo` to determine if the ethernet address in the BOOTP request is known. The `NetInfo` hierarchy is searched, starting with the local domain, looking for a `/machines` entry with an `en_address` property whose value matches the address of the broadcaster.
3. If the ethernet address is known, the server responds to the client with information including the path to the bootstrap loader. This path is comprised of two components: the directory, and the file name. The directory is determined from `/etc/bootptab` (yes, even if `NetInfo` is running); this is the `tftpboot root` directory. The file name was specified by the broadcaster (the client) in 1, above, as `boot`.
4. A request from the client to transfer the bootstrap loader is received; the loader is sent back, using TFTP.
5. Another BOOTP request is received, this time for the pathname to the kernel. The path is sent, determined from a combination of `NetInfo` and `/etc/bootptab`. Specifically, the `tftpboot root` directory (at the top of the tree from which kernels can be obtained) is found in `bootptab`, as is the default kernel to be sent. The kernel to be sent can be overridden in `NetInfo`: the `bootfile` property in the client's directory in the `/machines` directory specifies this. (Domains are searched on the server, starting with the server's local domain, continuing up the `NetInfo` domain hierarchy.) It can also be overridden by the client, in the boot command specified [e.g., `ben()new_mach`]; note that any explicit path given for a kernel will be relative to the `tftpboot root` directory.

Transfer of the kernel, using TFTP, completes this phase of the boot sequence.

One other note. This notion of `tftpboot root` is a bit unusual. A `chroot(2)` is only effected during the actual transfer. If a path such as `../MyMach` (e.g., `ben()../MyMach`) is given for the kernel, strange things will result. Specifically (assuming the normal `tftpboot root`) `/private/MyMach` (that is, `/private/tftpboot/../MyMach`) must exist, and `MyMach` must exist in the `tftpboot root` directory. It is the latter of these which will actually be TFTPed to the client. Note also that the kernel file name shouldn't have any dash characters (-) in it: the dash is used by the PROM's boot command to

denote arguments to be passed to the booted kernel.

Valid for 1.0, 2.0, 2.1, 2.2, 3.0, 3.1, 3.2, 3.3