25th March 1997

Engineering Support Application Note
Number: 298
*Issue: ***DRAFT****

Bootstrapping and Configuration of Ethernet-based NCs from UNIX Fileservers

This document describes how to configure a networked UNIX host to enable it to export the NC bootstrap sequence and enable Ethernet-based NCs to discover the requisite information to make use of it.

Applicable Hardware :	Ethernet-based NCs and UNIX fileservers	Related Application Notes:	None

Distribution: **CONFIDENTIAL**
Registered Developers ONLY

Engineering Support
Acorn Risc Technologies
Acorn House
645 Newmarket Road
Cambridge
CB5 8PB

Introduction

This document is a step-by-step guide to setting up and configuring an NC. It is assumed that the reader is conversant with:

- UNIX (BSD, SVR4, Solaris, SunOS, Linux... all these variants may be used)
- BOOTP
- NFS & PCNFS

When an NC is switched on, it looks for a BOOTP server in order to discover the following:

- NC's IP address
- IP address of the nearest router
- Subnet mask
- The directory that the NC should mount, containing the shared start-up files and application directories.

NB. Key applications are stored in the internal ROM, however, there is limited space in the there. Therefore software upgrades, and various other packages are held by the server to be requested on demand.

In order for the above procedure to be completed successfully, the server must be properly configured. This process is described in the following section.

Prerequisites

I. Suitably programmed smartcard obtained from Acorn Online Media (although for most application development work, this is not needed; see Appendix A)
Currently all smartcards are progammed with the PIN '1234'. This can be changed from the NC administration menu, once the system is up and running.

II. A UNIX 'tar' archive file containing the NC server software distribution.

Server Configuration

In order to configure the system, the administrator should log in as the super-user.

IP Addresses

The host name and IP address to be assigned to each NC should be added to the **`/etc/hosts`** file and/or NIS (Network Information Service, formerly YP) map. If NIS is used, it is important to force (or wait for) a push of the maps.

NC file system installation & modifications

A set of files is provided for the NC, and these should be copied into the appropriate directory on the server (e.g. **`/export/home/ncd`**).

As the files in the boot sub-directory are used as a shared resource by all the NCs connecting to the server, they should have their ownership set to that of the standard un-privileged network user, and their access permissions configured for read-only. Depending on which flavour of UNIX is being used, this user can be called "nouser" (e.g. SCO Unix 3.2v4.2) or "nobody" (e.g. BSD, Solaris, SunOS) this latter naming convention is the one used in this document. Please check your UNIX system manual for further details.

A complication which arises at this point is that the credentials for the un-privileged user map to different UID, GID tuples, depending on which version of UNIX is running. The Acorn NC considers "nobody" to have a UID of 32767 and GID of 9999; if your UNIX server disagrees with this, you may wish to add an additional UID and GID to `/etc/group` (on BSD systems) and the password database.

It is possible that when you examine your files under UNIX, they will appear to belong to "nobody", but the NC will request access with a different UID, GID pair. Therefore, to ensure that no ownership problems occur, type:

```
chown -R 32767:9999 ~boot/*
```

To remove write-access to the files, type:

```
chmod -R ugo-w ~boot/*
```

NB. The convention of using tilde ("~") to represent the default home root directory for the NC is used throughout this document; you should substitute the relevant absolute pathname.

Directories held in the `users` subdirectory can optionally belong to full UNIX users (with the appropriate ownership, username and password settings), or can be given the same anonymous UID and GID as the boot branch of the NC directory tree. If full UNIX users are created, this has security benefits but has the disadvantage of a higher administration overhead. The alternative is to give full control to all users of the system, which would not matter if only NCs were connecting to the server (the mount path of each user is fixed in the smartcard). However, non-NC users would have unrestricted access to these accounts in this case.

When configuring new users, it is recommended that the file structure exemplified by the demo user account be replicated in the home directories. Please consult your UNIX documentation for information regarding setting up new user accounts.

NB. There is no reason why existing user accounts cannot be utilised by the NC (e.g. `/export/home/dwalker`).

In the boot directory extracted from the server archive, the file `!NFSBoot/!RunEther,feb` requires the following changes to be made:

```
|| Ethernet NC Server Boot File
||
<.. cut ..>
|| Set up the name resolver.
||
If "<Inet$Resolvers>" = "" Then Set Inet$Resolvers 136.170.131.2 136.170.131.3 -Change to your local DNS hosts

Set Inet$LocalDomain acorn.co.uk                - Change to your local domain name
ResolverConfig
```

If you are not using DNS on your development server then the final three lines in the code fragment above can be commented out; but bear in mind that you will need to type in IP numbers rather than textual machine addresses when addressing non file:// URLs.

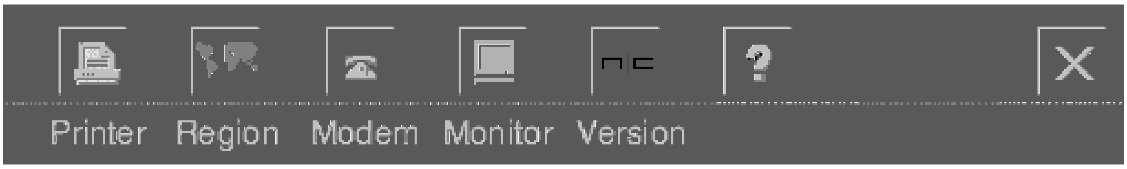

Figure 1 - The NC Administration Options menu bar

NFS Export

In order that the Network Computer can read files in the NC directory structure now set up (e.g. `/export/home/ncd`), it must be shared using NFS. Note that the `/export/home/ncd/boot` directory should be exported as a read-only filesystem, whereas the individual user directories should be exported read-write with the appropriate permissions to maintain access control.

In BSD systems, this is achieved by amending the `/etc/exports` file (see `man exports` for details). After making the changes to the NFS exports database, it will be necessary to prompt the controller daemon to re-read the file in order for the changes to take effect. You should be aware that some flavours of UNIX require any exported hierarchy to be a complete filesystem and not just a directory within a filesystem; unless you have `/export` (or whatever) stored as a separate partition (which gets mounted according to `/etc/fstab` at system boot time) you may need to specify all of `/` in `/etc/exports`. Some UNIX variants may also require the `-alldirs` suffix added to the export option; see `man exports` to determine whether this is the case on your server.

For SVR4 systems, use the share command to add new nfs shares (see `man share`), e.g.

```
share -F nfs -o ro /export/home/ncd/boot
```

You may wish to check for error messages after re-starting NFS (e.g. typing errors in the exports file). Typically, error messages are recorded in the system log (see `man syslog`), and the last few error messages can be viewed by typing:

```
tail /var/log/messages
```

PCNFS

If file authentication is to be used with the NC, then `pcnfsd` must be installed on the host server. This means that if you wish to authenticate users against their UNIX credentials, you should obtain a copy of pcnfsd. Several public domain sources exist, as well as commercial products (e.g. http://www.sun.com/solstice/Networking-products/PC-NFS51.html). As the NC uses the RPC support in pcnfsd rather than UDP, variants such as `rpc.pcnfsd` (from BSD 4.4 distributions) can be used with appropriate systems. See `man pcnfsd` to find the filename and location of your pcnfsd and how to start it.

BOOTP

1. BOOTP must be installed on the system to enable the NC to get it‘s configuration information from the server. The BOOTP server (`bootpd`) can normally be found in `/usr/etc` or `/usr/sbin` (for further details see `man bootpd`). If the software is not installed it will be possible to obtain it from your vendor. Alternatively there are various publicly available versions (see ftp://sunsite.doc.ic.ac.uk/computing/operating-systems/unix/FreeBSD/FreeBSD-stable/src/libexec/bootpd).

2. The file `/etc/services` (see `man services`) should contain the following lines:

```
bootps      67/udp      #Bootstrap Protocol Server
bootps      67/tcp
bootpc      68/udp      #Bootstrap Protocol Client
bootpc      68/tcp
```

3. The file `/etc/bootptab` needs to be modified to contain an entry for the NC:

```
            .default:\
            :ht=ether:\
            :bf=/export/home/ncd/boot:\
header      :ds=xxx.xxx.xxx.xxx:\
            :gw=xxx.xxx.xxx.xxx:\
            :hn:\
            :sm=xxx.xxx.xxx.xxx:\
            :vm=rfc1048

entry       ncxxx:tc=.default:ha=xxxxxxxxxxxx:ip=xxx.xxx.xxx.xxx
```

Figure 2 - Example bootptab template (don't type "header" and "entry" in)

The header section represents the various boot configurations. The header defines:

- ht - host hardware type
- bf - bootfile (The NC uses this field to pass the NFS mount path which holds the !NFSBoot application).
- ds - Domain Name Server address list (optional for development work)
- gw - gateway (network route) (optional for development work)
- sm - subnet mask
- vm - vendor magic cookie selector (vendor specific options)

Each entry comprises a name for the NC, followed by the following definitions:

- tc - table continuation (points to similar ‘template” host entry)
- ha - host hardware (Ethernet MAC) address
- ip - host IP address

Note that the following values should be added to bootptab (numbers refer to the annotations in Figure 2):

`:ds=xxx.xxx.xxx.xxx:\`	Replace underlined section with DNS IP address
`:gw=xxx.xxx.xxx.xxx:\`	Replace underlined section with router IP address
`:sm=xxx.xxx.xxx.xxx:\`	Replace underlined section with subnet mask
`ncxxx:`	Replace underlined section with name for a given NC (we suggest the use of "nc" followed by an ascending numeric)
`ha=xxxxxxxxxxxx:`	Replace underlined section with hardware (MAC) address of the NC
`ip=xxx.xxx.xxx.xxx`	NC IP address

NB. Some bootp daemons (e.g. SCO Unix 3.2v4.2) use slightly different bootptab file formats, and it is possible that the suggested configuration will cause the daemon to crash. If this occurs remove the leading full-stop (".") from the .default header and entry, then re-run bootpd. In addition, some bootptab files may already contain a .default entry which breaks the above model; if so, remove the pre-existing entry. Alternatively consult your UNIX system manual.

4. The file `/etc/inetd.conf` (BSD) or `/etc/inet/inetd.conf` (some SVR4) should include:

```
# <service>  <tli>   <proto> <flags> <user>      <server_pathname>        <args>
```

```
bootp        dgram    udp      wait    root        /usr/sbin/bootpd          bootpd
```

inetd is the internet "superserver" which should be run at boot time by `/etc/rc.local`. It‘s purpose is to listen for connections on certain internet sockets, decide what service the socket corresponds to, and invoke a program to service the incoming request.

Finding the NC’s hardware address

The MAC (Ethernet) address of each NC client is usually found on the outside of the box, or in a packing slip. Alternatively plug the NC into a suitable monitor (eg an AKF60) and the power supply; the MAC address (in the form xx:xx:xx:xx:xx:xx) is the first item displayed on screen.

If you are using a Domain Name Server, the **`arp`** command will show entries similar to below:

```
Device      IP Address              Mask              Flags              Phys Addr

le0         base-address.mcast.net  xxx.xxx.xxx.xxx                  01:01:01:01:01:01

le0         hostname                xxx.xxx.xxx.xxx                  02:02:02:02:02:02

le0         0.255.255.255           xxx.xxx.xxx.xxx                  00:40:b8:00:00:01
```

Find the devices that don’t have hostnames, or that have IP addresses not within your domain range. These devices need to be assigned an IP address, which should be added as an entry to the bootptab file along with the listed Physical Address (see Figure 2 - Example bootptab template).

NC Boot procedure

When the Ethernet NC is switched on, it goes through the following boot sequence:

1. NC broadcasts physical address in a bootp message

2. NC server bootp process recognises this message, and does a lookup in it’s bootptab to see if the hardware address is listed. If this is the case, it sends a bootp reply to the NC comprising the information illustrated in Figure 2.

3. The NC mounts the server at the point defined by the ‘bf’ field of the bootp reply; pcnfs is not used at this point as the NC mounts as user ‘nobody” (UID: 32767, GID: 9999), ie user authentication is not required.

4. The NC starts the !NFSBoot application, by executing the !Run,feb obey file. This detects that the NC network hardware is Ethernet, and switches control to the !RunEther,feb file (see Appendix A).

5. At the end of this process, the ‘insert smartcard” screen is displayed (see Figure 3).

6. User enters a four digit PIN which (if correct) unlocks the NFS mount details on the smartcard.

7. User home mount request is made. The following information is read from the smartcard:

·File server IP address (NB. This NFS file server must be running pcnfsd for authentication purposes.)
·Username
·Password

8. NC Browser requests the home page.
pcnfs is not used at this point as the NC mounts as user "nobody" (UID: 32767, GID: 9999)

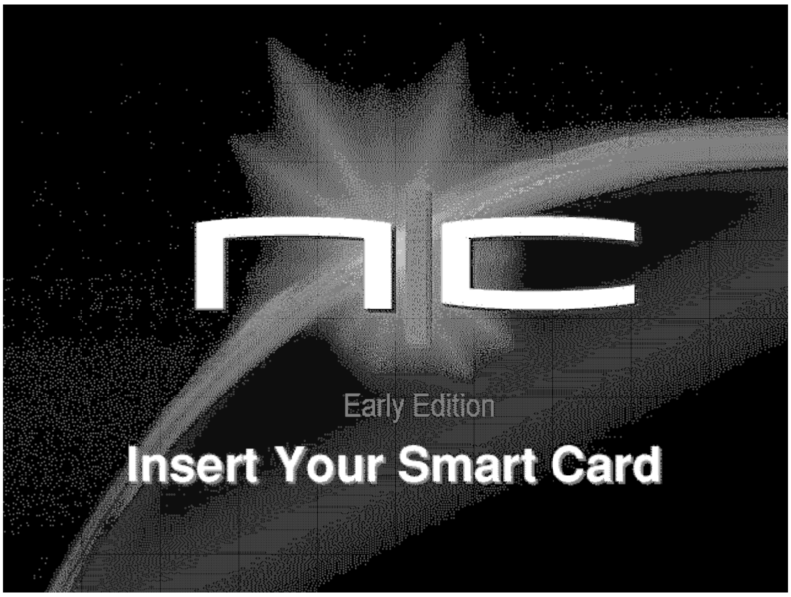

Figure 3: "Insert Smartcard" splash-screen

Circumventing Smartcard Authentication

For the purposes of software development, most developers will not need to utilise the user identification and authentication services offered by the Smartcard interface. It is possible to circumvent the Smartcard checking stage (and the splash-screen above) and "hardwire" the NetStation to start up at a specific user' s homepage by modifying the **~boot/!NFSBoot/!RunEther,feb** file as follows:

* Comment out the last line **Desktop System:Apps.!Sennen**
* Insert the following below the |**Desktop System:Apps.!Sennen** line:

NFS
logon <*username*> <*userpassword*>
Mount NCDHome /export/home/ncd/users/<*userdirname*> - or wherever the named user has their home directory
Set NCFresco$Home Home.html
Set NCFresco$Welcome Home.html
Resourcefs
Desktop NCFresco:!Run

This will now allow you to logon as the named user and allow you read-write access to the server; the NetStation will boot to your homepage if you have copied the files from **/export/home/ncd/users/demo** to the named user' s home directory.

For interactive development on the NC itself, yo may wish to insert **basic** instead of the lines above, following |**Desktop System:Apps.!Sennen** ; be aware that the NC will complain if you attempt to boot it straight to a command-line environment.

Problems and Solutions

Please read the following before requesting assistance.

The NC cannot find the BOOTP server, which is installed and running.

Ensure that the NC is connected to the network, and that the connection is live.
Test the connection by plugging the NC ethernet cable into a configured machine, and check that it is responding to network packets using the `ping` utility.

Make sure that the MAC (hardware) address of the NC‘s ethernet card is correctly entered into the BOOTP configuration file bootptab (see section IV.c)

I cannot find BOOTP on my system

It may not be provided as standard. You should obtain a copy.
Try searching the Internet for a public domain version eg. Solaris source code can be found at ftp.mc.com:/pub/bootp-2.4.1.tar.Z

Alternatively, contact your machine vendor for further details.

Bibliography

Introductory:
Washburn & Evans, ‘TCP/IP, Running a Successful Network”, Addison-Wesley
Hegering & Läpple, ‘Ethernet - Building a Communications Infrastructure”, Addison-Wesley

Intermediate/Advanced:
Stevens, W.Richard, “TCP/IP Illustrated Volume 3”, Addison-Wesley

Glossary

BOOTP - The BOOTstrap Protocol allows a diskless client (e.g. NC) to discover it‘s IP address, the address of a server host, and the name of a file to be loaded into memory and executed. After this address and filename information has been obtained, control passes to the second phase of the bootstrap where the file is transferred from the host to the client, and executed.

BSD - Berkeley Software Distribution-based version of UNIX (e.g. SunOS).

DNS - Domain Name Service. Distributed database system for resolving symbolic names to Internet addresses (i.e. IP addresses). e.g., om66.acorn.co.uk -> 136.170.135.116

HTML - HyperText Mark-up Language

NC - Network Computer.

NIS - Network Information Service (formerly yp). NIS‘ purpose is to provide information that is required throughout the network, to all machines on the network. Information that is likely to be distributed by NIS includes login names, passwords, home directories (`/etc/passwd`), and group information (`/etc/group`). So, for example, if your password entry is recorded in the NIS passwd database you will be able to log in to all machines on the network that run NIS client software.

NFS - The Network File System was developed by Sun Microsystems, and provides transparent remote access to shared file systems across networks. The NFS protocol is designed to be machine, operating system, network architecture, and transport protocol independent.

ROM - Read-Only Memory.

SVR4 - System 5 Release 4-based version of UNIX (e.g. Sun Solaris 2.x).

Appendix A: !RunEther,feb file

Note: Additional modifications that need to be applied to this file for Ethernet NCs with Modem ROM version numbers less than 0.26, are underlined. Comments regarding the changes that need to be made are given in square brackets.

```
|| Ethernet NC Server Boot File
||


|| Set paths as required.
||
Set Boot$Dir <Obey$Dir>
Set Boot$Path <Boot$Dir>.
Set System$Dir <Boot$Dir>
Set System$Path <System$Dir>.
Set Root$Dir <Boot$Dir>.^
Set Root$Path <Root$Dir>.
Set WWW$Path Resources:$.Apps.
Set WWWRoot$Path Resources:!Boot.HTML.Startup.
Set Images$Path Root:Images.
Set NCHelp$Path Root:Help.<TV$Type>.,<NCHelp$Path>
Set User$Path NFS::NCDHome.$.
Set NCWriter$SavePath User:Files.Word.


|| Set any network routes required (BOOTP can set default gateway).
||
|| eg. System:Utils.route -e add net default 136.170.129.77 1
```

[To add the network route, uncomment the above line (remove the `||`*) and replace the underlined IP address with the address of your router.]*

```
|| Set up the name resolver.
||
If "<Inet$Resolvers>" = "" Then Set Inet$Resolvers 136.170.131.2 136.170.131.3
```

[Replace the two underlined IP addresses with the IP addresses of the appropriate primary and secondary Domain Name Servers (respectively). If you are not using DNS, comment out this line and the following two lines.]

```
Set Inet$LocalDomain acorn.co.uk
ResolverConfig


|| Load the Client Authentication Module (only do this if your server is
|| running NCAS configured for your DES enabled smartcards). The system
|| variables override entries on the smartcard.
||
||Set NCAS$IPAddr <NCAS IP address>
||Set NCAS$ListenPort <NCAS listen port>
||Set NCAS$ConnPort <NCAS connect port>
||RMEnsure CAM 0.01 RMLoad System:Modules.CAM


|| Load the new format smartcard registry modules
||
RMEnsure NCRegistry 0.20 RMLoad System:Modules.NCRegistry -init
RMEnsure NCRegistrySpprt 0.20 RMLoad System:Modules.NCregSS


|| Load the server version module.
||
RMEnsure NCConfig 0.18 RMLoad System:Modules.NCServer


|| Set up SafeRun module to allow protected files to be run.
||
RMEnsure SafeRun 0.00 RMLoad System:Modules.SafeRun
SafeRun_Register NFS::NFSBoot.$.


|| Load the HTTP module which supports SSL (comment these lines out if you
|| don't want to load SSL).
||
If "<HTTP$SSLVersion>"<>"2" Then RMLoad System:Modules.httpsmod
RMEnsure HTTP 0.61 RMLoad System:Modules.httpsmod


|| Allow printer PDFs and resources to reside on server.
||
Set Printers$Path System:Printers.,<Printers$Path>
RMReinit PPrimer


|| Load boot block.
||
|| This provides settings which override smartcard contents eg. name server,
|| time server etc. which are local to the ethernet network. Only uncomment
|| this line if you have a valid boot block for your server.
||
|| Registry_Load System:Boot.netblock/scbb


|| Setup browser variables.
||
Set NCFresco$Welcome WWWRoot:InsertC
Set NCFresco$Home User:Home/html
Set NCFresco$Hotlist User:Choices.NCFresco.Hotlist
Set NCFresco$Cookies User:Choices.NCFresco.Cookies
Set NCFresco$Plugins User:Choices.NCFresco.Plugins
Set NCFresco$Users User:Choices.NCFresco.Users
Set NCFresco$Config User:Choices.NCFresco.Config


|| Set paths for applications.
||
Set Director$Path Root:Director.
Set Replay$Path Root:Replay.


|| Boot applications.
||
Run Replay:!ARMovie.!Boot
Run Replay:!MovieFS.!Boot
Run Director:!MMPlayer.!Boot


|| The following lines should only be uncommented if
|| plug-ins embedded in the browser are to be enabled.
||
RMEnsure WindowManager 3.87 RMLoad System:Modules.Wimp
RMEnsure Window 1.33 RMLoad System:Modules.Window
Run Director:!Shockwave.!Boot


|| Allow customisation.
||
IfThere Boot:Custom.<Inet$HostName>.!Run Then Run Boot:Custom.<Inet$HostName>.!Run


|| Start up the UI for the NCD
||
Desktop System:Apps.!Sennen
```