

RPL.DOC

**LAN Manager Remoteboot Service
Standards and Procedures**

**As implemented at the University of South Australia
by the Information Technology Unit**

(Dan Shearer - See version history - Appendix A)

Contents

1 Introduction.....	3
1.1 This Document and Normal Remoteboot Administration.....	3
1.2 What is Remoteboot?.....	3
2 Setting Up and Customising Remoteboot.....	3
2.1 Previous Installations.....	3
2.2 Installing Remoteboot Files.....	4
2.3 Initial Customisation.....	5
2.3.1 General Items.....	5
2.3.2 RPL Tree.....	6
2.3.3 Changes to server configuration.....	7
2.4 Campus-specific Problems and Customisations.....	8
2.5 How to clone a RIPL server.....	8
2.6 Dealing with new card types and/or address prefixes.....	9
3 How a workstation remoteboots.....	9
3.1 When machine is turned on.....	9
3.2 Server response.....	9
4 Tuning Workstation Configuration Files.....	11
4.1 RPL.MAP.....	11
4.2 BBLOCK\NDIS\ELNK*\DOSBB.CNF.....	12
4.3 DOS500.FIT.....	13
4.4 CONFIG.SYS.....	15
4.5 AUTOEXEC.BAT.....	15
4.6 LOGOUT.BAT.....	16
5 Running RPLMGR.....	16
6 Typical Problems.....	16
6.1 Hangs at Waiting for Data Frame during initial load.....	16
6.2 Tries to start as wrong card.....	16
6.3 Message Error in Server.....	17
6.4 Workstation won't start.....	17
6.5 Logs on, but doesn't connect to any servers.....	17
6.6 Logon scripts don't work.....	17
6.7 DOS Packet Driver software doesn't work.....	18
6.8 Message Bad or Missing Command Interpreter.....	18
6.9 NET START WORK Message Runtime error R6009.....	18
6.10 RPLMGR Message SYS0002: Cannot find the specified file.....	18
6.11 Boots slowly, then hangs.....	18
6.12 Command NET VIEW doesn't work.....	19
6.13 Error message: Error in server \\servername (System files not found).....	19
6.14 Error message: NOT USER LEVEL SEC!.....	19
6.15 Error message: Cannot write to specified device.....	19
6.16 Error message: Error in server \\servername NetBIOS AddName failed.....	19
6.17 ELNK3 hangs after rplbind2 with no error message.....	19
6.18 Message Error 2603 while assigning perm. to remoteboot dirs.....	20
6.19 Message Error 2245 while assigning perm. to remoteboot dirs.....	20
6.20 Message Password shorter than required in RPLMGR.....	20
6.21 Error 2586 occurred while assigning permissions to remoteboot subdirectories.....	20
7 Hardware Considerations.....	20
8 Remoteboot Security.....	20
9 Problems with the Remoteboot Service.....	21
10 Testing Remoteboot.....	22
Appendix A - Version history for this document.....	24
Appendix B- Diagram of RPL tree.....	25

1 Introduction

1.1 This Document and Normal Remoteboot Administration

This document, RPL.DOC, contains notes on the LAN Manager Remoteboot service as used at UniSA. Administrators will still need to refer to the LAN Manager Administrator's Guide for general information. RPL.DOC is maintained in the UniSA CCLIB shared document area as RPL.DOC. A related document is the text file RPL.TXT, held in the same place. RPL.TXT records where all changes are made to the remoteboot service, but not what they were. Another relevant file is in LMPARAM.DOC which contains the UniSA recommendations for tuning servers, including RIPL servers.

The UniSA customisation of remoteboot contains many optimisations that have been worked out over time, and care should be taken when changing anything at all.

A simple procedure is available for safe modification and testing of those parts of the service which are most likely need it (see section 10 *Testing Remoteboot*) but changing anything else is likely to cause trouble unless it is carefully coordinated. Normally, administrators of remotebooted workstations will be able to make customisations needed using the usual methods, which are independent of the method used for booting.

1.2 What is Remoteboot?

In this document, remoteboot is the art of making a DOS-based computer think that a network connection is a C: drive. The computer may or may not have an internal hard drive. The point is that every request for information from drive C: comes from the network, even to the level of boot sectors.

LAN Manager remoteboot is a very complete solution using unusual programming techniques. It is a hack in the best traditions of computer trickery. The latest version is for LAN Manager 2.2b however UniSA is unlikely to upgrade to this since it involves paying money. The current version (2.1a) is remarkably reliable.

2 Setting Up and Customising Remoteboot

2.1 Previous Installations

If a remoteboot installation has ever been attempted on a server for a previous version of LAN Manager, installing version 2.1a could have unexpected side-effects. Problems we have experienced include old versions of the Nokia/Microsoft OS/2 COMTKR driver not being overwritten, and RIPL modules in LANMANDRIVERS\PROTOCOLRPL such as ACSLAN.DLL not being installed at all. The safest procedure is to check for an existing or attempted previous installation by looking for the files mentioned, and deleting them if they exist. Unfortunately there is no routine to completely remove RIPL from a server. Another possible trap is the existence of a RPLFILES sharename, no matter where it points. RPLINST returns a

silly error message and the installation aborts. Delete the sharename and start again. (See section 9 *Problems*, point 13.)

2.2 Installing Remoteboot Files

(This section is a modification of the 2.1a release file README.RPL.)

1. Run SETUP.EXE from the LAN Manager root directory.
2. Choose Configure.
3. Choose Network Drivers.
4. Choose <Add/Remove Protocols>.
5. Add Remoteboot. If it is already there (perhaps because of an aborted previous installation) then remove it and immediately re-add it without exiting SETUP, making sure that you have first deleted the directory tree D\$\LANMAN\RPL. If you are deleting the tree remotely using a DOS utility you may have to link to a sharename to delete some of the trees under RPLFILES separately, because the directory nesting is too deep for DOS.
6. Select <OK>, then <OK>, and then <Save>.
7. Insert the Microsoft LAN Manager OS/2 Drivers disk when prompted. The necessary files will be copied, more than 200 of them. If the discs were created from images stored on a server you may not have all the files, again because the directory structure is deeper than DOS can manage, when an extra level or two is added through being itself put in a subdirectory. Note also that if copies are held on a server somewhere you can happily install RIPL over the network, which is a much quicker procedure. (At UniSA you can link to \\LVSERV4\LMSET to install these files. OS/2 doesn't have a directory depth limit as low as DOS so this will work from the OS/2 SETUP program.)
8. After the files have been copied, exit the Setup program.
- 8.5 Edit C:\CONFIG.SYS and check the /CACHE parameter on HPFS386. It has probably been reset to AUTO by SETUP, which is deadly (see LMPARAM.DO.) Reset this to 3000, or whatever it was before.
9. Shutdown the server and reboot the server, to ensure the remoteboot driver is loaded. (If the server is in use, you can delay this step until you have to do a NET START REMOTEBOOT.)
10. Run RPLSETUP. At UniSA, link a drive to \\LVSERV4\DOSRPL and specify that as the drive for RPLSETUP to copy files from, to save time. Otherwise you will \\need to insert the LAN Manager DOS workstation discs when prompted.
11. Copy DOS files. All files from the C:\DOS directory of a normal DOS 5 installation need to be copied to (RPLFILES)\BINFILES\DOS500. As well, the system files IO.SYS and MSDOS.SYS need to be present, without attributes. This copying must

be done manually for (paranoid) copyright reasons, but you should find the files in one of the RIPL servers already installed. There will already be a file called BOOTSECT.COM present in the DOS500 directory which should not be touched.

12. Run RPLINST. This checks that all files are present, sets access permissions and sharenames if needed, creates initial profiles and UAS groups. RPLINST will *not* run if the sharename RPLFILES points to a non-existent place. You cannot delete the sharename and then run RPLINST either; you must restart the server first. (See section 9 *Problems*, point 13.) RPLINST may do things other than listed above, see sections 6.13 and 9.14. It is RPLINST that adds *remoteboot* to the *srvservices* line in LANMAN.INI. This will probably take a very long time. Stop every service except for SERVER and WORKSTATION before doing this (especially NETLOGON. They seem to interact and make the thing take even longer.)
13. Replace the RPL\BBLOCK\NDIS\ELNK16.DOS driver if it is version 2.01. This version does not work with Windows 3.1, later versions may do and version 1.1 certainly does. Version 2.01 is dated 28/8/92 and is 9792 bytes in size and is the version that comes with Lan Manager 2.1a.
14. Reboot the server to load COMTOKR and other essential drivers so that the RIPL service will start. It will have already been added to the *srvservices* line of LANMAN.INI by setup.

By this stage you should have a working remoteboot server, although it will not be very useful. Test to see if it is working, with all card types. You should be able to get a prompt, login/out and do manual network connections. Make sure you do this on a network that does not already have a remoteboot server of some kind on it, or there will be a conflict.

2.3 Initial Customisation

2.3.1 General Items

This section explains how to get a default remoteboot installation into a form that can be used to boot a default UniSA pool. After each major modification you should test to see if everything still works. There are a lot of things that can go wrong.

1. Create a new profile for each of the cards you will be using. At UniSA this will normally be Etherlink, Etherlink II and Etherlink III cards only (DECPA hardware is not supported as yet.) Use RPLMGR, and select Remoteboot/New Profile for each. You cannot edit RPL.MAP directly when first adding a profile, although direct modifications to an existing entry are possible. When you do edit RPL.MAP, make sure RPLMGR is not running or your changes may get overwritten. Read **Rule Number One** in section 4, *Tuning Workstation Configuration Files*. Then, edit RPL.MAP and for each profile definition change the bootblock name to suit the kind of card that pool has: RDOSL for Etherlink, RDOST for Etherlink II (leave the default RDOS for Etherlink 16s.)
2. Add workstations to these profiles. Lists of ethernet addresses for all pools are available, so obtain the latest list and add them using RPLMGR's Remoteboot/New Workstation option. After reading Rule Number One again, edit RPL.MAP and go to the bottom, where new workstation records are added. For each workstation

change field 3 to 'D'. This flags a default boot, without prompting for a workstation name or password. All UniSA remoteboot are default. Try reading section 6 if you get an obscure error.

3. Do *not* make an entry in the password field in the New Workstation dialogue box. It refers to a remoteboot password, which is quite different from the password associated with a Lan Manager account. It can be confusing if something is inadvertently put in this box, since for a workstation that has the prompting field set to 'N' or 'D' this message appears at the next boot:

```
RPLDISK.SYS:
Error in server '\\LVSERV1'.
Server not responding : "login"
```

and the 'bad password' count is incremented at the RIPL server. Note that an identical set of messages appears if RPL.MAP is edited manually to refer to a user for the first time (ie one that is not a member of RPLUSER group, among other things.) Whatever you enter as the 'Computer Name' will be both the computer name and default login name when that adaptor boots. A normal Lan Manager password is never prompted for. Although it works, this is a security hole (See section 7, *Remoteboot Security*). The above error message can also occur when there is a password on an account. Even if an account has *Password required : No* and *User may change password : No* it seems a user is permitted to put in a password that has no effect anywhere except for remoteboot, where it causes the above error in a 'D' booting account. ***The preceding problem appears to have been fixed as of LAN Manager 2.1a but more testing is needed.***

2.3.2 RPL Tree

Next you need to customise the RIPL tree. It is assumed that files to be copied come from an existing UniSA RIPL server.

1. backup DOS500.FIT in the (RPLFILES)\FITS directory and copy the UniSA DOS500.FIT.
2. create the directory (BINFILES)\UNISA and copy its contents
3. create the directory (PROFILES)\GENERIC and copy its contents. Replace the .SYS files with configs suitable for the particular pools this server supports. Unless there is special hardware in a pool (eg real-time cards or CD-ROM drives) these should be just the same as other pools with the same network cards. At present the only difference between pools is the EMM386 line, which varies with card type. Don't forget to copy the GENERIC\LANMAN.DOS subdirectory containing LANMAN.INI.
4. UniSA does not use BINR to save FIT memory and fiddling with paths. Copy RPLINIT.EXE and RPLPRO2.COM from (BINFILES)\BINR to (BINFILES)\LANMAN.DOS\SERVICES
5. For security and to save space, delete all unnecessary files in (BINFILES)\LANMAN.DOS and NETPROG, such as NETADMIN.EXE, SETUP.*, LMSETUP.EXE and MSD.EXE. (See section 6, point 18)
6. Copy (BINFILES)\LANMAN.DOS\NETWKSTA.500 to NETPROG*.EXE. This is because we assume DOS 5 for simplicity and to save a FIT entry. It can be overridden with a FIT entry for some other operating system, if need arises.
7. Make the NDIS-to-Packet driver converter available. Copy DIS_PKT.DOS to (RPLFILES)\BBLOCK\NDIS. You should be able to just copy the .CNF files from an existing installation, but if not then modify the DOSBB.CNF files in the

(RPLFILES)\BBLOCK\NETBEUI\do leave a record of what changes you have made in the comments section. Modify the PROTOCOL.INI files in these directories also, to include a [PKTDRV] section. If you get this last bit wrong the boot will work, but RPLBIND2 will report 'Status=33' and the packet driver will not be installed. See Section 4 for more information on DOSBB.CNF.

8. Disable the post-boot profile creation. Taking the same care as in the previous step, comment out the RPLPRO1 line in DOSBB.CNF (you will not need to do this if you have just copied the .CNF file from an existing setup.) Remove RPLPRO2 from the WRKSERVICES= line in the generic LANMAN.INI, ie (PROFILES)\GENERIC\LANMAN.DOS\LANMAN.INI. This step avoids the need for a writable temporary area when remotebooting, and saves about 3k of memory. Section 3 has details on what the post-boot profile is.

Although there are some directories and files for cards, networks and operating systems not relevant to your installation they should be left. They do not take up much space, and can be sensitive - for instance, even on an all-ethernet installation removing the token-ring directories can cause trouble. The RPLFILES tree is not so delicate - see sections 2.4 and 6.10.

2.3.3 Changes to server configuration

See the UniSA document LMPARAM.DOC, which contains recommendations for tuning servers, including RIPL servers.

Threads

The number of OS/2 threads is set in C:\CONFIG.SYS and should be at least 188 (the OS/2 default is 128.) Of these, up to *maxthreads* are used by the remoteboot service, where *maxthreads* is set in the servers' LANMAN.INI. One thread is used per workstation boot request, and when *maxthreads* is exceeded the results are unpredictable but involve a hang during the initial stages of a boot. *Maxthreads* = 27 seems to be reasonable at UniSA, in instances where there are no more than 120 workstations on the same campus, and there are 2 RIPL servers per campus, and not more than 40 people want to boot at any one time.

NCBS, SESSIONS

These two parameters need to be set quite high for a heavily loaded server, and a UniSA remoteboot is often under load. You can find out what the current best recommendation is by looking at LMPARAM.DOC. The symptoms of these being too low are remoteboot workstations booting slowly or not at all, and at the same time other workstations observe the server to appear and disappear from view on the network.

[COMTOKR_XIF]

The following tuning parameters in the server's PROTOCOL.INI make remotebooting more reliable, particularly through several bridges or with old cards. There are still

failures to maintain a connection at times on a very heavily loaded network, but this is reasonable. We'd rather a remoteboot failed than an existing user lost their connection!

```
T1_TICK_ONE=7
T1_TICK_TWO=38
T2_TICK_ONE=3
T2_TICK_TWO=15
TI_TICK_ONE=38
TI_TICK_TWO=149
```

2.4 Campus-specific Problems and Customisations

There may still be reasons why remoteboot doesn't work on a particular campus. Among the more common of these are:

- No script file defined or present for each pool user's account. See section 3.2, near the end, for a brief description of how logon scripts are used at UniSA. Remember that the logon scripts must be present in the NETLOGON share for every server in a domain that is running the NETLOGON service (at UniSA, this is likely to be all servers.)
- Hardware might not be up to remoteboot. 80286-based machines do not seem able to cope very well, and Etherlink I cards can be very difficult depending on network conditions. At least 1Mb of workstation memory is needed to provide a reasonable amount of conventional DOS memory after booting.

Various customisations can be made, and many of the files and directories installed can be deleted. See 6.10 for a warning on how to do this.

2.5 How to clone a RIPL server

Hopefully all the RIPL servers will eventually be replicated from a single master, since there are only 3 files that differ between campuses (RPL.MAP, RPLMGR.INI, LANMAN.DOS.) Occasions might arise where a RIPL server has to be set up quickly as a clone of another. To do this follow steps 1 to 9 of section 2.2, and then

1. Using the OS/2 file manager, delete the LANMAN\RPL tree (you can also use `DELETE -drsy LANMAN\RPL`.)
2. From a Windows NT machine, `XCOPY` the RPL tree from the source server to the empty LANMAN\RPL on the new remoteboot server.
4. Run `RPLSETUP`, specifying N (No) to all options. This should give the message "updating domain entries" and terminate after a few seconds.
3. Run `RPLINST`. Check any errors against Section 6.
4. `NET START RPL` and test that workstations of all card types are logging in. Also run `RPLMGR` and try to add a new workstation.
5. Check the server tuning entries in Section 2.3.3

If you try to add a workstation using `RPLMGR` in the new setup and it complains, the `/E` parameter was probably missed from the `XCOPY` command which skips some profile-related empty directories. If `RPLINST` refuses to run check that it runs on the source server, then check that you specified `/E` on `XCOPY`. If `RPLSETUP` refuses to run check the `RPLFILES` sharename. Check Section 6 for other error messages.

As yet we haven't got REPL going, but the public domain program EQTREE is likely to achieve the same results with much less trouble if applied carefully.

2.6 Dealing with new card types and/or address prefixes

This is covered in the manual in chapter 13, **Booting Workstations Remotely**, however it doesn't mention how to add another ethernet address prefix for an existing card type. This is documented in the comments in RPLMGR.INI: use "|" as a separator in the configuration sections to indicate ranges of card addresses. The same thing must be done in RPL.MAP in the boot block fields.

3 How a workstation remoteboots

3.1 When machine is turned on

When a pool computer is turned on, the 3Com EPROMs in all UniSA remotebooting workstations prior to Etherlink IIIs display a line something like:

```
Version 1.0 Copyright 3Com Corp. 1987
```

and then broadcast an XNS bootblock request. If there is no remoteboot server working on the network, there is some network fault (such as an ethernet cable not plugged in) or there is some workstation hardware fault (eg network card address in the wrong part of memory) then the EPROM will display the error message:

```
Fail to load bootpc.com, check cable!
```

If there is only a 3+Share 3Start service running you will see a UniSA banner and be prompted for a pool name. Etherlink III PROMs (from the Canadian company Lanworks) differ from the above because they no longer use XNS, instead they use IBM's DLC protocol. (DLC is very similar to NetBEUI.) After the initial bootstrap has occurred they behave just like the other cards.

3.2 Server response

The RIPL service running on a LAN Manager server on the network will notice the requests for bootpc.com (or the DLC equivalent, for Etherlink III cards) and send it down. Bootpc displays the following screen while it tries to get all the code and data needed for booting:

```
Searching for RIPL Server, Retries = 000000  
Sending File Requests to RIPL Server, Retries = 0000  
Waiting for Data Frame with Sequence Number: 000000
```

A busy network or busy RIPL server can cause one or two retries in the second counter display, but there should not normally be more than this except in the case of the very slow Etherlink I cards. The first time a server has an Etherlink I (3C501) card boot from it the boot will certainly fail, but subsequent requests from the same or any other 3C501 will usually work. (This may be due to a DLL not being loaded after the REMOTEBOOT service has been started following a system reboot, and will also

occasionally happen after a long time without a 3C501 booting - perhaps the .DLL has been swapped out.) The server looks at the ethernet address and either

1. if this card is not in the RPL.MAP database, it determines what brand of card is being booted from the vendor code in the ethernet address (first 6 bytes) and sends it the most likely bootblock as defined in RPLMGR.INI, or,
2. finds the ethernet address in RPL.MAP and sends it the right bootblock according to the profile the workstation is joined to.

The process described in point 1 above is of no use at UniSA because it does not give adequate configuration management control.

Before the bootblock is sent, all FIT file macro translations are performed. See the heading **DOS500.FIT** in Section 4 for more details about FIT files. Once bootpc has loaded all the server has to send (about 90 to 110 data frames, depending on the card, ie on the size of the *.DOS driver) it transfers control to RPLBOOT.SYS and then you will see a line some thing like:

```
MS DOS LAN Manager Protocol Manager v2.0
```

followed by other lines of network-related installation information. Next there is a line similar to:

```
*** UNISA MC36 ETHERLINK II REMOTEBOOT CONFIGURATION
```

followed by normal CONFIG.SYS information as on a standalone workstation. This information comes from a COMMENT.SYS line in CONFIG.SYS. It is important to recognise the difference between the programs above and below the ***UniSA line, because the source files are controlled by quite different mechanisms, DOSBB.CNF before the stars and CONFIG.SYS after them. As soon as CONFIG.SYS has been executed a RPL-specific AUTOEXEC.BAT is executed, which first displays a UniSA banner asking the user to wait while the automatic logon (ie NET START WORKSTATION) is done, which takes some time. The /FIT and /RPL qualifiers on the NET START mean it also does a NET LOGON. /FIT allows the .FIT-defined (CNAME) to be used as the /COMPUTERNAME: name, so there is no need to type in any workstation identification. The macro (CNAME) is defined by the values in the RPL.MAP database used by RPLMGR (see section 4.) Since a logon script is defined for all UniSA accounts this is run during the last stage of the logon and the various environment variables (eg %HOMESERVER%) are set by means of SETENV /ALL commands in the scripts. The (GENERIC)\AUTOEXEC.BAT is specific to remoteboot workstations and is in addition to the usual AUTOEXEC.BAT on the home server. It sets some essential things (eg the variable %LH%) and then links to \\%HMSERVER%\PUBLIC and runs the AUTOEXEC found there.

The thing that NET START WORK /RPL normally does is to create a PROTOCOL.INI in a writeable place. This is only for those who need to dynamically load and unload protocols, especially TCP/IP. Since this is not relevant to UniSA student machines, we have stopped it by the last step of the customisation procedure outlined in Section 2. The major benefit of this is that we do not introduce another potential security hole, although there are small memory and speed savings.

The program RPLLINK in AUTOEXEC may not be of use at UniSA, but its purpose is to relink the remoteboot drives back into the DOS driver chain so that they are visible

again as devices, eg PROTMAN\$. Some programs need to open these things as files, such as the dynamic loading and unloading of the protocols TCP/IP, DLC, XNS and IPX. There is no memory penalty so it has been left.

In the event of something going wrong during the boot process after the bootstrap program is loaded but before CONFIG.SYS is executed, the workstation often reboots by itself, following a timeout period. This is a useful feature in case of temporary network problems or the RIPL server being restarted, but it can be a bit surprising at times. It is not consistent, however and appears to depend somewhat on the kind of ROM being used although the ROM is not involved after the initial bootstrap load.

4 Tuning Workstation Configuration Files

Rule Number One: if you are changing any configuration file in the RIPL tree you must leave a comment in the history section as to what you have done and why. All files discussed in this section have a comment section, so please use it.

As delivered from Microsoft, the RIPL files are in an enormous and cumbersome directory tree under the LANMAN directory, although this has been considerably improved with version 2.1a. This is the most general possible solution, able to individually configure workstation boot configurations for any combination of a large number of kinds of network cards, operating systems, protocols and client machines. As suggested in the manual we have customised the setup. The UniSA pools differ from many environments in that they are fairly homogeneous:

- All users are have the same security level (priv=User, no password)
- All workstations run the same, single primary protocol (NETBEUI) and no secondary protocol
- There are only four kinds of remoteboot network cards (Etherlink I, II, 16, and III)
- Only ethernet
- Only ISA workstation architecture
- Only DOS, and only version 5

Server files under D\$\LANMAN relating to different parts of the bootstrap process are:

Detecting cards, loading bootpc.com	..\NETLIB\RPLNET*.DLL
Boot block (RPLBOOT.SYS, network drivers)	BBLOCK\NDIS\ELNK*\DOSBB.CNF
Network part of CONFIG.SYS (built by RPL)	BBLOCK\FITS\DOS500.FIT
Non-network part of CONFIG.SYS (used as-is)	RPLFILES\PROFILES\GENERIC*.SYS
First AUTOEXEC.BAT (RPL-specific)	RPLFILES\PROFILES\GENERIC\AUTOEXEC.BAT

The remoteboot server has a RPLROOT sharename pointing to D\$\LANMAN\RPL, and the tree below it is shown in Appendix A.

4.1 RPL.MAP

This will need occasional manual modification.

This file is the database the remoteboot service and can be manipulated using RPLMGR, an OS/2 Presentation Manager utility for administering remoteboot workstations. RPLMGR is concerned with the maintenance of the directory tree under

the RPLFILES\MACHINES, CONFIGS, TMPFILES and PROFILES directories, and it also has pointers to files in RPL\FITS and RPL\BBLOCK. The format of RPL.MAP is specified in comments within the file. The 2.1a *Update for Administrators* details the changes in record formats. The parts of interest to UniSA are the definitions of:

- Bootblocks: what cards they are associated with, what files they are in and how they are downloaded.
- Profiles: what bootblock workstations in this profile use.
- Workstations: what ethernet address a particular username is associated with, what profile it belongs to, what File Information Table they use.

Normally this information is set up once and not touched; however if an ethernet address is changed or added, or a new pool (=profile) is added then this can be done either by editing RPL.MAP or modifying workstation properties in RPLMGR. Ensure that RPLMGR is not running when you edit RPL.MAP and preferably stop the REMOTEBOOT service although this is not necessary for editing most of the record types. Changes in RPL.MAP generally take effect after the next RIPL request.

4.2 BBLOCK\NDIS\ELNK*\DOSBB.CNF

These files should only be modified in the most exceptional circumstances. Even putting a space in the wrong place can cause havoc. Microsoft's remoteboot specialist <randyn@microsoft.com> approves of the UniSA setup but adds the warning:

Sorry we don't document dosbb.cnf, rpl.map, rplmgr.ini, etc very well, but there is a reason for that. RIPL is a rather difficult system to administer; we try to build in as much flexibility as the user will need, but we don't extend that to forming custom boot blocks etc. It is extremely easy for a user to try to modify dosbb.cnf, blow it, and mess up his RIPL server badly.

If they are modified, remember that the REMOTEBOOT service must re-read the bootblocks every time there is a RIPL request following a change and that this does not always happen correctly. It is very likely to fail if machines are booting from the server at the time of a change. If there is a problem, stopping and starting the RIPL service should fix it. At times the RIPL services notices changes and stops itself, which is a good idea except no notification is sent of the event, although an entry is made in the error log. At other times the error log entry is made, stating that RIPL has been stopped, although in fact it never is stopped.

There is a different DOSBB.CNF for each kind of card booted. It contains information on which bootpc.com should be sent to a booting workstation, and how it should be sent. It specifies driver names and amounts of memory needed, and other obscure parameters which are not documented by Microsoft. These are the files run before the normal CONFIG.SYS (the code which is downloaded when the "Waiting for Data Frame" numbers are counting.) The only UniSA customisations have been to insert the NDIS-to-TCP packet converter, and remove RPLPRO1. DIS_PKT.DOS is needed by programs which use the TCP/IP protocols (such as NCSA Telnet) which must be run here or it will not work at all.

Although there is no official documentation for .CNF files, Randy did volunteer the following:

On .cnf files: they are processed from bottom to top, so the last driver listed is the first one installed. Following the driver name are:

1. parms, if any (with tilde standing in for embedded spaces)
2. a small number giving extra memory if the program needs it
3. 'M' if the program is moveable after it initializes. M is a very dangerous switch, and I would recommend you not use it unless you are very brave. In general, I can't guarantee a program is 'clean' in this sense unless I have tested it myself. since drivers loaded in the boot block are loaded in memory from the top down (as opposed to from the bottom up as the dos would normally load them), drivers are inherently not as memory efficient when loaded during rpl. 'M' can recover some memory, if the driver was written in a 'clean' (memory position-independent) fashion.

See comments on the file CONFIG.SYS in this section for details on the usual method of installing device drivers etc.

4.3 DOS500.FIT

This file is unlikely to need modification. A mistake can stop all or some machines from booting, with very odd error messages.

A File Information Table is a hardware independent method of deciding what files will appear on the remotebooted workstation's virtual C:\ volume, and which files they are mapped to in the RIPL directory tree. It is a very useful facility, and is the main way in which the UniSA setup has been able to be made so similar for all pools, without compromising on performance or making testing difficult. The only documentation available on how .FIT files work is an article in the Microsoft Networking Journal by Randy Nevin, March/April 1992 Vol. 3 No. 2. This article is helpful but covers a more general case than we have at UniSA.

The format for a FIT entry is this:

template replacement

where *replacement* is the desired file name in the remoteboot volume (eg C:\SOGGY.EXE, for the utility SOGGY) and *template* is the source in the RIPL tree, starting at the directory D\$\LANMAN\RPL\RPLFILES (eg BINFILES\FREDUTIL\SOGGY.EXE.) Macros are allowed, which define the parameters that differentiate profiles and machines. The form of a macro is

(*macroname*)

where allowable macros are:

PROFILE	Profile name of the workstation, as in RPL.MAP (eg L318E16)
CNAME	Computer name, also from RPL.MAP (eg LIB01)
RPLFILES	Path for RPLFILES share (usually D:\LANMAN\RPL\RPLFILES)
BINFILES	Relative to (RPLFILES), if BINFILES sharename not defined
TMPFILES	Relative to (RPLFILES), if TMPFILES sharename not defined

Be careful when using (BINFILES) and (TMPFILES). If you leave the brackets out they almost-work in some circumstances, because they are also directory names. This can be quite confusing. Also note that there is not a (PROFILES) macro, and if you use it or any other non-existent name there will be no error but the file or directory will not exist (and cannot subsequently be seen on that workstation.)

The following very useful facts have been found to hold through all testing so far, and should be read in conjunction with the FIT translation algorithm that Randy provided in the MSNJ article:

- 1 if a file has a mapping entry but that file does not exist in the RIPL tree no error is generated, the target file just does not exist on the workstation's virtual C: drive. The exception to this seems to be when C:\LANMAN.DOS\LANMAN.INI is defined twice, with the first instance pointing to a non-existent file. In this case LANMAN.INI never exists.
- 2 each .FIT entry takes up about .2k of Upper Memory Block area.
- 3 no memory is used by a .FIT entry pointing to a non-existent file.
- 4 if there are multiple entries defining the same replacement with different template files, the longest template takes effect, counting only the path length not the filename. Others rules are discarded and do not consume any memory. The exception is LANMAN.DOS\LANMAN.INI, as specified above, which can only be defined once under any circumstances. This is probably a bug. (We now know it is, because Randy says so. It is also sometimes unreliable.)
- 5 If a .FIT line has a non-existent source it must be overridden by another line or that target file will never exist on the virtual C: drive (even if a file of that name is implicitly present by being in the (PROFILE) directory.)
- 6 A file named implicitly with a directory location is not the same as a file named explicitly by a pathname. In other words, DIR <directory> won't show the file, but DIR <directory>\<filename> will!

Each remotely booted workstation has a FIT file translation and its translator loaded into memory at boot time, and every time there is a request to access the virtual drive C: the request is routed through the translator to turn it into UNC (Universal Naming Convention) form. Unfortunately there is no way to easily determine the UNC mappings for a workstation profile without booting it and laboriously checking by hand.

Comments in a .FIT are preceded by a semicolon. The UniSA DOS500.FIT is very different to the standard one, and following is Version 1.02 of the DOS500.FIT (not including version documentation) with explanations in italics:

(RPLFILES) *This must be present, for unknown reasons*

;default directory

C:\ PROFILES\GENERIC*All files under GENERIC will be seen in C:*

;Machine-specific testing files - no effect if target files do not exist

C:\DEBUG.RPL PROFILES\GENERIC\CNAME).DBG

ie if TMC36.DBG exists under \GENERIC, then the workstation TMC36 will have C:\DEBUG.RPL existing when it boots. All other workstations can see only TMC36.DBG. The RIPL AUTOEXEC.BAT sets the %DEBUG% variable if DEBUG.RPL exists, so if an administrator wishes to turn debug on as early as possible in the boot process all that has to be done is create a file.

C:\CONFIG.SYS PROFILES\GENERIC\CNAME).SYS

if a CONFIG.SYS exists for this particular workstation, then it takes precedence. This is highly useful for testing. Otherwise, this line has no effect

C:\LANMAN.DOS\LANMAN.INI PROFILES\GENERIC\LANMAN.DOS\CNAME).INI

this is similar reasoning, only for LANMAN.INI. This is unlikely to be used a lot, allows maximum flexibility

C:\AUTOTEST.BAT PROFILES\GENERIC\CNAME).AUT

This should hardly ever need to be used, but the facility is there. (GENERIC)\AUTOEXEC.BAT always checks for the presence of this file and chains to it if it is present. This is consistent with the existing UniSA standard

;Profile-specific files

C:\CONFIG.SYS PROFILES\GENERIC\PROFILE).SYS

This is the normal CONFIG.SYS; one per profile (=pool) and only one line differing in each, the EMM386 line. It could be extended to have only one line's difference per card-type, but this does not allow for unusual hardware to be put in a particular pool which requires special memory handling or device drivers.

;Machine-specific files

C:\LANMAN.DOS\PROTOCOL.INI (TMPFILES)\LOGON\CNAME).PRO

For some reason NET START WORKSTATION /RPL /FIT needs to write a whole new PROTOCOL.INI. These files all go in one place giving us as a useful byproduct a log of when a workstation was last started, since we have replaced (CNAME)\PROTOCOL.INI by (CNAME).PRO. Another advantage of this is that when workstations need to be added or deleted the only change is to delete a line in RPL.MAP - there is no associated directory or permissions needed.

;Shared DOS files

C:\IO.SYS (BINFILES)\DOS500\IO.SYS

C:\MSDOS.SYS (BINFILES)\DOS500\MSDOS.SYS

C:\BOOTSECT.COM (BINFILES)\DOS500\BOOTSECT.COM

C:\DOS (BINFILES)\DOS500

C:\UNISA (BINFILES)\UNISA

;Shared DOS LANMAN files

C:\LANMAN.DOS (BINFILES)\LANMAN.DOS

C:\LANMAN.DOS\LANMAN.INI PROFILES\GENERIC\LANMAN.DOS\LANMAN.INI

There is a different FIT for each operating system that can be remotebooted, at least. Individual profiles or even individual workstations can have different .FITs, but as yet UniSA has only needed the single DOS 5 one because of the flexibility the variables allow.

4.4 CONFIG.SYS

This is the only file which is not the same for all remotedbooted machines in the normal UniSA scenario. As shown in the commented DOS500.FIT above, there is a different CONFIG.SYS for each profile defined. This is because the EMM386 UMB-provider needs to have areas of RAM excluded that are used by hardware so that it does not allow the allocation of UMB space to these areas (see the DOS manual and Windows Resource Toolkit for further explanation.) The various pools have different hardware requirements, and rather than exclude large areas of UMB space and catering for all hardware configurations it was decided spend the extra effort to tune a number of CONFIG.SYS files.

As usual, care is needed when working with CONFIG.SYS files since inappropriate EMM386 exclusion and inclusion parameters can hang the machine. The settings already calculated for particular card types are likely to be optimal, anyway.

4.5 AUTOEXEC.BAT

This is a development of the Microsoft default batch file, the main unusual part being to do with environment variables. The login script run by the NET START WORK /RPL has SETENV /ALL commands that set variables specific to the pool, such as the

name of the home server. After the NET START WORK has been done the workstation name is put in an environment variable using the SETENV /COMPNAME. The RPLLINK command is described in section 3. If there are errors, such as the workstation's home server not being found or the workstation not starting, AUTOEXEC displays an error message and locks the keyboard for 10 minutes, before automatically rebooting the machine (in the hope that the problem has gone away, whatever it was.)

4.6 LOGOUT.BAT

This is the only batch file in the PUBLIC area on the servers which has remoteboot-specific code. It allows a user to logout and login again with a different username, so that for instance a staff member first boots up a pool machine remotely, then exits from the menu and runs this batch file. It detects whether the C: drive is actually a remote device using CHECKSYS (CHECKDRV doesn't work on Lan Manager remoteboot C: drives) and if so sets some variables and calls the batch file C:\NET\LOGIN. This batch file should be identical to the one on staff workstation hard drives, and it must be kept in step when modifications are made.

5 Running RPLMGR

[this section unfinished]

Brief overview of the features of RPLMGR useful at UniSA

6 Typical Problems

6.1 Hangs at *Waiting for Data Frame* during initial load

Possible causes:

1. The card is almost certainly an Etherlink I card. You may have to reboot three or four times to get one of these to boot, however, once it has booted there should be no further problems. Cold boots are usually less successful than warm ones. If it happens with any other type of card then there is either a hardware problem with this particular card (try running the diagnostics) or there is a network problem such as congestion. Try another machine with the same hardware configuration.
2. Too many machines are trying to boot at once. The number of OS/2 threads is specified in the C:\CONFIG.SYS file on the server. See section 2.1.3, and the UniSA document LMPARAM.DOC.

6.2 Tries to start as wrong card

If after the initial program load sequence there is a message such as *3Com 3Startion Driver Version nnnn* when the hardware is an Etherlink I card, or something like *3Com Etherlink 16 driver* when you know there is an Etherlink II/16 installed, the boot will always fail since the software driver does not match the hardware. This is usually

because a card has not been defined in the RPL.MAP database with RIPL manager, or has been defined incorrectly. Check the ethernet address and which profile it has been added to. (See Section 5 - *Running RPLMGR*) The server will attempt to start a machine that is not in the database by guessing the card type, and is often wrong. The card must be added to the database anyway even if it guesses correctly, so make it a habit to add all cards straight away. Profile-specific parameters are needed to boot and you do not get these without being in the RIPL database.

6.3 Message *Error in Server*

Possible causes:

1. Uncommon, but there really could be some error in the remoteboot server. It is sometimes possible to confuse the RIPL service by doing manual updates to RPL.MAP at the wrong time, or modifying bootblock items. Stop and restart the service and try again.
2. Wrong ethernet driver associated with this card in RPL.MAP, or a corrupt driver is present.

See also 6.13, and especially 2.1.3, point 3

6.4 Workstation won't start

The NET START WORKSTATION command is unusual in that it also logs the user on when used with a /RPL parameter. Possible causes of failure are:

1. A password is accidentally specified in RPL.MAP as well as the 'D' default-boot parameter. This will certainly cause a hang. See section 2.3.1, point 3.
2. Server is running out of resources. See section 2.3.3.

6.5 Logs on, but doesn't connect to any servers

At UniSA all linking is done via environment variables which specify the server to link to. At boot time the variable HMSERVER is set (among others) in the logon script, as explained in section 4.5. If the workstation complains that it cannot link to anything but servers are visible with a NET VIEW (from a workstation *not* started with RIPL - see 6.12) check the environment with a SET command. If HMSERVER is missing, then either the logon script doesn't exist or it is not being run. Check that the script does exist on both the primary and backup servers for this domain and then see the next section.

6.6 Logon scripts don't work

1. No logon script defined for this user.
2. The workstation is in the wrong domain and so it can't see a logon server that has the correct logon script. It is possible that it does have an account for this workstation (through a coincidence of names) and so a script will be run, but not the correct one. This most commonly arises through copying RIPL tree files from

another site. Change the default domain specified in LANMAN.INI in the directory (PROFILES)\GENERIC\LANMAN.DOS

3. Not enough environment space available. Remove some variables, or increase the amount specified in the CONFIG.SYS for this profile. May be card-dependant, certainly varies with some other hardware specifications.

6.7 DOS Packet Driver software doesn't work

Look carefully at the messages that go past while booting. RPLBIND2 or PROTMAN may complain, in which case see section 2.3.2 point 7 for information on setting up packet drivers with remoteboot. Check that all the PROTOCOL.INIs have been changed for packet drivers (one per card-type.)

It may be a problem with the software itself. If you have copied it from another segment of the network or it has never been installed, check that the NETMASK and RARP entries are present and correct.

6.8 Message *Bad or Missing Command Interpreter*

The most common cause for this is forgetting to provide a (PROFILE).SYS in the GENERIC directory, ie the workstation will not have a CONFIG.SYS.

6.9 NET START WORK Message *Runtime error R6009*

Followed by:

*Not enough space for environment
NET 3546: Error running C:\LANMAN.DOS\SERVICES\RPLINIT.EXE*

This happens on Etherlink I cards that have their memory buffer set to an invalid address (In our case, E000 or more rather than C800.)

6.10 RPLMGR Message *SYS0002: Cannot find the specified file*

There are lots of files that can be deleted in a RIPL installation, at UniSA typically about 1000 can be deleted per server. However, RPLMGR will refuse to add machines to a profile if the directory (PROFILES)\(PROFILE)\WKSTA.PRO does not exist, and probably the file DOSSHELL.INI must be present in this directory as well. All other files in the (PROFILE) and (MACHINES) and (TMPFILES) trees can be deleted if they are not specifically referred to.

6.11 Boots slowly, then hangs

Booting slowly can be caused by a busy network, but it can also be due to incorrect allocation of resources on the remoteboot server. Booting slowly and then hanging is almost certainly a resource problem. See section 2.1.3. Sometimes the slow Etherlink I cards will be a bit overwhelmed by a very busy network, and will boot very slowly but still work.

6.12 Command *NET VIEW* doesn't work

This is correct. Remoteboot workstations have *numviewedservers* set to 0 in their LANMAN.INI for a measure of security and also to save a little memory.

6.13 Error message: *Error in server \\servername (System files not found)*

Happens if

1. you try to clone a RIPL server by overwriting the RPLFILES tree without both running RPLINST again and (this is sometimes needed) going into RPLMGR and then exiting.
2. you add a user to RPL.MAP manually, and forget to add him to the RPLUSERS group with NET GROUP. RPLINST sets permissions and puts users in the RPLUSERS group.

6.14 Error message: *NOT USER LEVEL SEC!*

This message appears after the RPLBIND2 banner has appeared, and at UniSA occurred when the RPL server was ill and not able to take any more NetBIOS connections. The condition can be many things though. Randy says:

the message NOT USER LEVEL SEC comes from rpldisk.sys, and it is printed out when rpldisk determines that the server it is trying to establish a session with is not running user level security, which means it is share level. however, its also kind of a catch all, since it can get printed out for a variety of errors, basically things that keep the session from getting established correctly. unknown username? server process not started? various network hardware problems? it is printed after the NEGOTIATE PROTOCOL smb is exchanged, if the encryption key size in the response smb is zero, meaning that something is wrong. either its not running user level security, or something else interfered with the successful completion of the negotiate protocol

6.15 Error message: *Cannot write to specified device*

This appears when doing a NET USE. It just means that it is trying to append to or create LMUSER.INI on an unwriteable drive. Change the parameter SAVECONNECTIONS in LANMAN.INI to NO.

6.16 Error message: *Error in server \\servername NetBIOS AddName failed*

This is just what it says, ie an AddName NetBIOS API call failed. The most common reason for this is having another remoteboot card booted on the network with exactly the same name.

6.17 ELNK3 hangs after rplbind2 with no error message

Check to see if there is a hard disc defined in CMOS. Even though an ELNK3 is supposed to offer a choice (*Press H to boot from hard disc*) with some machines it doesn't. Change CMOS to have no drives defined (having first written down the drive type!)

6.18 Message *Error 2603 while assigning perm. to remoteboot dirs*

Running RPLINST while not logged on with Admin privs gives this message. So does something else, not sure what yet.

6.19 Message *Error 2245 while assigning perm. to remoteboot dirs*

Running RPLINST after cloning a RPL installation gives this message after doing lots of work. Remotebooting should appear to work perfectly though, and RPLMGR will also run. Don't worry about it :-)

6.20 Message *Password shorter than required in RPLMGR*

You are trying to add a passwordless user with RPLMGR in a domain that has a minimum password length set. If you must do this you first remove the restriction, add the user and restore the restriction.

6.21 *Error 2586 occurred while assigning permissions to remoteboot subdirectories*

If you get this then you are not logged in with privilege. It might also mean that some files are missing in the RPL tree, but not too sure about this.

7 Hardware Considerations

[this section unfinished]

ROM settings for cards, Interrupts, etc

8 Remoteboot Security

To maintain security remoteboot assumes

1. only authorised persons have access to RPLMGR (reasonable)
2. the physical security of cards assigned privileged workstation names (dubious)
3. nobody can fake an ethernet address (dubious)
4. nobody will ever get the opportunity to increase the privilege of an existing remoteboot user (reasonable)
5. no RPLUSERS will read files in the default RIPL tree, eg RPL.MAP (ridiculous)

Furthermore, it would certainly be possible to determine the protocol by which a remoteboot workstation is logged in without prompting for a password, so that presumably a generic program could be written to fool the UAS from any machine connected to any LAN Manager network. See point 8 in section 9 - **Problems with the**

Remoteboot Service. *Except that this last problem seems to have been fixed from version 2.1a. More testing needed.*

9 Problems with the Remoteboot Service

1. Cannot define C:\LANMAN.DOS\LANMAN.INI more than once in DOS500.FIT. This seems to be related to the directory name, since I cannot reproduce it even in a directory call FRED.TOM. Maybe this is because LANMAN.INI is needed before CONFIG.SYS is run (hard to believe.) Maybe it is to do with the fact that LANMAN.DOS already has a default mapping (to BINFILES) and overwriting LANMAN.DOS\LANMAN.INI with yet another rule might be one rule too many. See section 4.3 points 1 and 4
2. Problems with ELNK cards. Even with all settings at the default ELNK cards still do not boot reliably. They often hang at the *Waiting for Data Frame* stage. Something further needs tweaking in the DLLs, although ELNK support is better than it used to be. Making the COMTOKR timeouts much longer helps. ELNK cards in faster machines tend to hang during boot more often than slower machines. Note that Lanworks claim in part of their ELNKIII manual that they have RPL PROMS for ELNK cards, so this might be a (far too expensive) solution. They also avoid claiming they make these PROMS in another part of the manual so who knows.
3. Cannot support TCP/IP as primary or only protocol, it must have NetBEUI as well. We are like many sites and would like to go to solely TCP/IP as soon as the stack became small enough to make it possible.
4. NetBEUI is statically loaded and bound. Would be good to be able to unload NetBEUI, even if we must have loaded first at boot time.
5. Rewrites workstation's protocol.ini every boot by default. Eventually discovered that RPLPRO1 is to blame - if you want to run a completely tight installation just get rid of this file. It is not acceptable to have the (TMPFILES) tree as per default in a hostile student environment.
6. No documentation for later than version 2.1.
7. Need to incorporate MSJ FIT article by Randy into documentation.
8. There seems to be some mechanism in RIPL to bypass logon security altogether. That is, I can define a card to be associated with a privileged account in RPLMGR and without any changes to the UAS or the server it will log itself on automatically at bootup, via the special /RPL /FIT features of NET START WORKSTATION. After testing this, the password in the privileged account will always be null. This implies some sort of protocol which allows the password to be deleted without human intervention. I am convinced that someone could debug the protocol and therefore presumably log on to any account on any LAN Manager network, if they wrote software that used this protocol. I have captured the packets involved in this exchange and will analyse them at some stage. The appropriate people in Microsoft have been notified. ***But the problem appears to have gone away as of version 2.1a. Further testing needed.***
9. Very obscure message if remoteboot password is set on a non-prompting w/s. See section 2.3.1, point 3.
10. Can't put * after NET START RDR (ie can't force password entry at logon. See)
11. There is a RPLBIND2 error message: Status=33. It occurs if [PKTDRV] wrong in PROTOCOL. This means what? What other numbers are there? This needs to be documented.

12. If there are too many REMs at the beginning of a remoteboot CONFIG.SYS it seems that RPLDISK.SYS has problems. This is not consistent but it happens fairly frequently if there are more than 20 or so lines of Rems. The sign is that there is a long pause after RPLPRO2 is run. Probably cause is that RPLDISK.SYS is running out of memory. This is sometimes hard to reproduce, however as a general guide keep comments down. This is somewhat card-dependant. (Have never been able to get it to happen on an ELNK16, for instance.)
13. RPLINST fails with an obscure message if a RPLFILES sharename already exists and points to an invalid spot. It may also fail if it just exists, doesn't matter where it points - not so sure about this second one though.
14. What is the procedure for turning a non-replicated RIPL setup into a replicated one without starting from scratch? See section 6.13
15. RPLMGR insists on the WKSTA.PRO directory in the profile directory. See 6.10 for details.
16. There is no indication on a workstation's startup screen as to which server has answered its RIPL request. This would be very helpful for debugging purposes.
17. Bad version of ELNK16.DOS shipped with 2.1a. See section 2.2, near the end.
18. Bad security management to make all utilities available to users. See Section 2.3.2.
19. Cannot delete the C: remoteboot connection, even if (as at UniSA) we have no further use for it after boot time. It would help greatly if we could delete the connection or at least link something else over the top of it (I don't know what the redirector would think of that.)
20. ?? NET START WORK /RPL /FIT doesn't seem to always return the same error message as a normal NET STARK WORK if the logon script wasn't run. This is not fully confirmed as yet. Futher testing on the general LM login script bug should make it certain.
21. Auto reboot on boot hang not consistent. See last para of 3.2. It would be nice if one of the TSRs did reliably detect a hung state and retry after a certain period.
22. RIPL often notices changes to *.CNF and halts (see section 4.2) but there is no alert message sent!
23. Should be program that semi-automates the procedure for adding a new card, especially the bits that involve editing RPL.MAP and RPLMGR.INI
24. Include in the documentaion details from Lanworks on how to add a new card, or their equivalent. Nice clear instructions.
25. RPLMGR sometimes refuses to recognise that an adaptor is not being used, so it won't allow it to be modified. The only way around this is to exit RPLMGR, and sometimes a STOP/START RPL needs to be done.

10 Testing Remoteboot

Because of the design of the UniSA remoteboot setup it is possible to do testing at different levels. It is possible to

1. Have a CONFIG.SYS for a particular profile. This is the usual situation, and is done with an entry in DOS500.FIT.
2. Have a CONFIG.SYS for individual workstations within a profile. This is also done with DOS500.FIT. An individual CONFIG takes precedence over a profile's CONFIG due to the FIT algorithm.
3. Turn DEBUG on for individual workstations. If the file <computername>.DBG exists in the GENERIC directory then the environment variable DEBUG will be set

when that machine boots as the test for DEBUG.RPL in AUTOEXEC.BAT then succeeds.

For general testing of a LAN Manager network at UniSA see the file STRESS.DOC for details.

Appendix A - Version history for this document

Ver	Date	Who	Comments
3.3	30 Mar 95	DPS	More info on cloning - esp. check /CACHE:AUTO after running SETUP. Always use Windows NT XCOPY. Added section 6.21. Check 2.3.3 when cloning.
3.2	3 Mar 94	DPS	General tidy-up before sending to Microsoft again
3.1	18 Feb 94	DPS	More minor points about cloning. More on funny error messages 2245. RPLMGR 'password too short'.
3.1	16 Feb 94	DPS	Changed How to Clone section. Added errors 2603 and 2245.
3.0	7 Feb 94	DPS	Document PROTOCOL.INI tuning in 2.3.2
2.9	4 Aug 93	DPS	More info about the error message in 2.3.1, section 3
2.8	29 Jul 93	DPS	Added 6.17
2.7	15 Jul 93	DPS	Added references in 6.3
2.6	12 Jul 93	DPS	Update logon script information Update 9.8
2.5	9 Jul 93	DPS	Added to 6.13 for RPLUSER group stuff
2.4	8 July 93	DPS	Add 6.15, 6.16
2.3	31 Mar 93	DPS	Add section on cloning a RIPL server Modify for some of Randy's comments. Rest later.
2.2	12 Mar 93	DPS	Spelling check. RPL >> RIPL to avoid confusion
2.1	3 Mar 93	DPS	Add more errors in 6.*. Tidy up. Add bugs in 9.*
2.0	24 Feb 93	DPS	Add some error details. Fix details SWS found.
1.9	12 Feb 93	DPS	Add bits about too many REMs in CONFIG.SYS Add about NCBS etc in server params
1.8	8 Feb 93	DPS	Add section "Typical Problems". Proofread throughout.
1.7	2 Feb 93	DPS	Add new information about FIT algorithm
1.6	26 Jan 93	DPS	Correct theory in 'How a W/S remoteboots'
1.5	8 Jan 93	DPS	Add RPLLINK Describe LOGOUT procedure
1.4	7 Jan 93	DPS	Incorporate more info from Randy Update for DOSBB.CNF changes Delete info on USERNAME.BAT Update FIT rules
1.3	6 Jan 93	DPS	Update (TMPFILES)\LOGON standard Describe problems with .FIT files
1.2	14 Dec 92	DPS	More 2.1a update
1.1	5 Nov 92	DPS	LM 2.1a update Better explanation of .FIT files
1.0	29 Aug 92	DPS	Original for LM 2.1

Appendix B- Diagram of RPL tree

Diagram of directory tree under the RIPL sharename of a UniSA remoteboot server. Many Microsoft-supplied directories are either not shown or have been deleted.

+---BBLOCK	Bootblock definitions and files. Hardware specific.
+---NDIS	- NDIS Packet drivers. DIS_PKT NDIS-TCP packet driver
+---NETBEUI	- RIPL bootstrap files, Protocol Manager, PROTOCOL.INI
+---ELNK	NETBEUI files etc. DOSBB.CNF lives here.
+---ELNK16	
+---ELNKII	
+---FITS	File Information Tables. UniSA uses only DOS500.FIT
+---RPLFILES	
+---BINFILES	All non-hardware-specific executables needed in the boot process
+---DOS500	- DOS 5 files
+---LANMAN.DOS	- LAN Manager files
+---DRIVERS	
+---DOSUTILS	
+---NIF	
+---PROTMAN	
+---XIF	
+---NETPROG	
+---NETWKSTA	
+---SERVICES	
+---UNISA	Files needed for UniSA remoteboot not supplied by Microsoft
+---CONFIGS	Not used by UniSA
+---MACHINES	One subdir for every workstation. Created by RPLMGR. Not used by UniSA so it has been deleted
+---PROFILES	One subdir for every profile. Created by RPLMGR. The only directory used by UniSA is GENERIC, with DEFAULT for use by cards which have not been defined by RPLMGR.
+---GENERIC	
+---LANMAN.DOS	
+---DEFAULT	
+---TMPFILES	Machine writeable files created during remoteboot go in the LOGON directory. This contains a last-booted record (section ??)
+---LOGON	
+---DEFAULT	