

AiR-BOOT User Manual

Version 1.1.4
(April 2017)

ArcaOS Edition

Please note:

AiR-BOOT accesses hard disks at the BIOS level.
When USB media are present, the handling of these media
strongly depends on the quality of the BIOS.

This version of AiR-BOOT is GPT and LUKS aware
and such disks are skipped to prevent them from appearing
in the main menu.

>> Reading this Manual before using AiR-BOOT would be a smart thing to do <<

Booting multiple operating-systems on today's machines is no trivial process.
While the utmost care has been taken to ensure the proper operation of this version of the
AiR-BOOT Boot Manager, nasty things can happen, so backup important data
before installing and using AiR-BOOT.

Please take the time to read the
[Before you begin](#), [Trouble Shooting](#), [Caution](#) and [Disclaimer](#)
sections in this manual.

Author : Ben Rietbroek
E-mail : rousseau.os2dev@gmx.com

Preface

This manual is about the AiR-BOOT Boot Manager as distributed with ArcaOS, the new OS/2 distribution.

The main purpose of AiR-BOOT is to be a replacement for the legacy OS/2 Boot Manager, which has several issues on modern computer systems, such as handling the co-existence of modern Windows and OS/2 on the same system and the use of hard disks with a capacity larger than ca. 512GB.

This manual is organized into sections that cover relevant topics about AiR-BOOT.

It starts with a short introduction of the beast after which the focus shifts to basics like:

Backup, Installing, Configuring, and Removing.

After that, some advanced usage is covered followed by a section on trouble shooting in case things do not work the way they should.

Please also read the caution section that covers installing recent Windows versions, as you may need to take some precautions to prevent them from wiping OS/2 systems installed in primary partitions.

What is AiR-BOOT ?

AiR-BOOT is a Boot Manager that can be compared to boot-managers like OS/2-BMGR, GRUB and others.

Like OS/2-BMGR and GRUB, AiR-BOOT presents a list of operating-systems from which the user can select one to boot the system. The interesting thing about AiR-BOOT is that it works on-the-fly, which means that it detects almost all changes to your disks and partitions without requiring your intervention. Another pleasant feature about AiR-BOOT is that it contains a built-in SETUP, allowing configuration from within AiR-BOOT itself, and yet is small enough to fit in the first track of the disk, (usually 63 sectors ~30kB in size), thus not requiring a separate partition.

Actually, it is one of the smartest boot-managers around for it's size, I kid you not !

A brief history

AiR-BOOT was developed by Martin Kiewitz more than 15 years ago, and over that time it matured into what it is now. It is completely written in Assembler and that's why it's so small in size, yet powerful enough to contain it's own SETUP together with a load of handy features.

Martin however, does not develop AiR-BOOT anymore and the last version by his hand is v1.06, as hosted on source-forge.

Kindly, Martin gave his permission to develop AiR-BOOT for further use with OS/2, a gesture which is highly appreciated. This manual is about the post v1.06 versions that are specially enhanced for use with OS/2 on more modern hardware. While the initial focus is to replace OS/2-BMGR, the longer term path is to enhance AiR-BOOT further to cope with upcoming things like EFI/GPT and other future developments.

AiR-BOOT as a replacement for the OS/2 Boot Manager

On modern systems, the OS/2 Boot Manager can have several issues.

One of them is its inability to boot modern Windows systems alongside ArcaOS. These modern Windows systems can use an extra primary partition for its boot-loader, and when OS/2-BMGR boots a system from a primary partition, it hides the other primary partitions. This causes the Windows boot-loader to fail.

Other issues with OS/2-BMGR are the handling of huge hard-disks and the requirement of a dedicated primary partition.

Huge hard-disks are disks with a capacity larger than 512GB. When such disks are used with OS/2, it uses an extended geometry to enable addressing beyond the 512GB boundary. OS/2-BMGR is not able to handle this extended geometry and will fail to display the boot-menu. OS/2-BMGR also requires its own dedicated primary partition, which can prevent creating more advanced disk layouts because in the MBR partitioning scheme, only 4 primary partitions can be created.

AiR-BOOT solves these issues:

- ◆ It does not require a primary partition
- ◆ It does not hide other primary partitions, so ArcaOS can be used alongside Windows OEM installations
- ◆ Huge hard-disks are handled correctly
- ◆ It comes with the SET(A)BOOT program that mimics the corresponding program for OS/2-BMGR
- ◆ It is open-source so adjustments and enhancements can be made

Some differences between AiR-BOOT and OS/2-BMGR:

- ◆ Partition and Volume labels are currently limited to 11 characters
- ◆ AiR-BOOT has its own configuration -- it does not use or take over any OS/2-BMGR configuration
- ◆ AiR-BOOT has a built-in SETUP which can be accessed by pressing F10

Before you begin

This section is about things to be aware of and things to do before installing AiR-BOOT.

AiR-BOOT installs itself at the start of the disk and modifies the Master Boot Record (MBR).

Besides elementary boot-code, the MBR also contains the Partition Tables that provide access to the data areas on the disk. A power outage or accidental reset of the computer during the installation of AiR-BOOT are examples of events that could cause these tables to get lost. Another example would be a severe bug in AiR-BOOT.

So it's important that you make a backup of the MBR to a file before installing AiR-BOOT. Store this file on a diskette or a usb-stick or whatever medium you have access to. Don't store it only on the boot-disk because you cannot access this disk (easily) in case of such an event. Now don't get scared, installation takes less than a second and the chances of ending up with a corrupted MBR are neglectable. But it's always better to be safe than sorry !

The DFSANOS2 disk-utility

On the ArcaOS Installation Medium, there is a disk-utility called DFSANOS2.

It can be found in the ?:\SYS\INSTALL directory, where ?: indicates the drive-letter of the ArcaOS Installation Medium.

This utility can be used to accomplish more advanced tasks with regard to OS/2 disk management and it is used in the section below to make a backup of the MBR. Before using this utility you should become acquainted with it on some test-system and always make sure you know exactly what you want to accomplish and what commands to feed it. For regular disk management tasks the graphical volume management programs should be used.

Backup the MBR and TRACK0

Below are some procedures to backup the MBR and TRACK0.

These procedures are not guaranteed to work on all systems and you may have to fiddle with the commands to get them working on your system. The objective here is to provide basic means to backup the above mentioned disk areas with common tools. There are many other ways to accomplish these tasks, like the DFSee program from FSys Software or the System Rescue CD available on the Internet.

The backups of the MBR and TRACK0 are specific to the point in time on which they are made.

This especially goes for the partition-tables in the MBR, which will change when disk modifications are made at a later date.

NEVER RESTORE AN MBR-BACKUP OR TRACK0-BACKUP AFTER DISK MODIFICATIONS HAVE BEEN MADE

OS/2

- ◆ Boot from the ArcaOS Installation Medium
- ◆ Start System Management
- ◆ Attach a medium like a floppy or (FAT16 formatted) usb-stick
- ◆ Open a Command Prompt

Give the following commands:

```
set PATH=?:\SYS\INSTALL;%PATH% <ENTER> (Where ?: is the ArcaOS Medium)
?: <ENTER> (Where ?: is the FAT16 Medium)
cd \ <ENTER>
DFSANOS2 image MBR.IMG 0 1 <ENTER>
DFSANOS2 image TRACK0.IMG 0 63 <ENTER>
?: <ENTER> (Where ?: is the ArcaOS Medium)
eject ?: <ENTER> (Where ?: is the FAT16 Medium)
```

Linux

- ◆ Download some Live-DVD like SystemRescueCD
- ◆ Mount the FAT16 Medium
- ◆ Open a terminal

Give the following commands:

```
sudo su <ENTER>
mount -l <ENTER> (Find the device-name of the Fat16 Medium)
cd /path/where/FAT16-Medium/is/mounted <ENTER>
dd if=/dev/sda of=MBR.IMG bs=512 count=1 <ENTER>
dd if=/dev/sda of=TRACK0.IMG bs=512 count=63 <ENTER>
cd / <ENTER>
umount /dev/device-name-of-FAT16-Medium <ENTER>
```

Windows

- ◆ Find out what version of Windows you are using
- ◆ Use Google to find information on the Internet on how to backup the MBR under Windows
- ◆ Do not run any downloaded Windows programs unless you have made sure they are safe

A brief Compatibility Check

- ◆ The hard-disk needs to have at least 63 sectors per track
- ◆ TRACK0 must not be used by some other disk management software
- ◆ The BIOS needs to support INT13X
- ◆ If the firmware is UEFI, full CSM support is required

However, if your system uses some form of non-OS2 LVM-software, or other disk related software that uses space in TRACK0, then AiR-BOOT should not be used. Also, GPT or otherwise none-MBR scheme partitioned disks are not supported at this time.

The Basics

This section covers the installation of AiR-BOOT and the basic things you need to know on how to use it.

A common scenario would be replacing OS/2 Boot Manager with AiR-BOOT, so that is the scenario used in the following sections. A sample configuration with OS/2 Boot Manager is presented in which OS/2-BMGR will be replaced by AiR-BOOT.

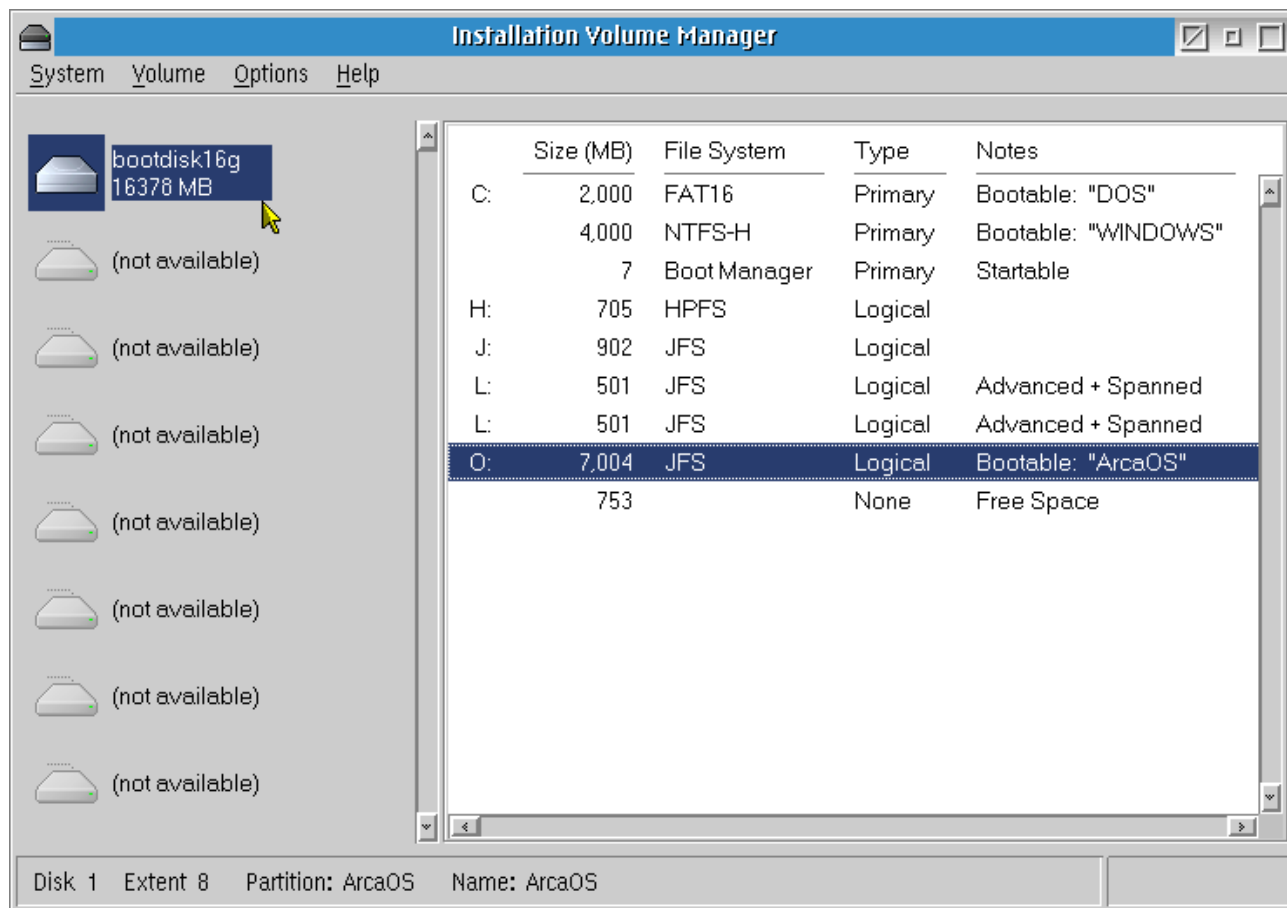
After this scenario, which of course covers installing AiR-BOOT, topics like operating and configuring AiR-BOOT pass the review. Finally, it is explained how to remove AiR-BOOT, and how to upgrade or downgrade it.

AiR-BOOT has some advanced features, like booting different OS/2 installations from the same drive-letter and chain-loading OS/2-BMGR, which are covered in the [Advanced Usage](#) section.

A Sample Configuration

Since AiR-BOOT was introduced as a replacement for OS/2 Boot Manager, it makes sense to use a sample configuration of multiple OS/2 installations that were installed using OS/2 Boot Manager. To further put some "flesh" on the configuration, a PC-DOS7 partition is present, multiple data-partitions and an LVM volume spanning multiple partitions. We will then install AiR-BOOT to replace OS/2 Boot Manager, see if everything still works and what the differences are. Needless to say that the disk was partitioned using the LVM Tools and that proper LVM Partition and Volume Labels have been set.

The Sample Disk Layout



The above screen-shot was taken before installing AiR-BOOT, with OS/2-BMGR active.

Short description of the Sample Layout

C: Partition "DOS"

This is a traditional DOS-Partition using the FAT16 file-system.

It was booted once with OS/2-BMGR to show its hiding behavior, which is reflected on the WINDOWS partition described below.

?: Partition "WINDOWS"

This is a traditional Windows-Partition using the NTFS file-system.

The -H suffix after the NTFS identifier in the File System column indicates that this partition is currently hidden. This hiding was caused by OS/2-BMGR which hides all other primary partitions except the one booted. This behavior of OS/2-BMGR is one of the reasons why it cannot be used with Windows Vista/7/8, because these systems use a boot-loader partition in addition to their system partition. When one of them is hidden, Windows fails to boot. AiR-BOOT will always unhide all partitions unless you explicitly configure otherwise. More information about this is in the section about configuring AiR-BOOT.

Boot Manager

This is the OS/2 v4.5 Boot Manager.

We are going to replace it's functionality with AiR-BOOT. Note that installing AiR-BOOT does not remove or cripple OS/2-BMGR, it just disables it by taking over the boot-code in the MBR. That's why AiR-BOOT is able to chain-load OS/2-BMGR.

(Caution when doing that !)

H: Partition

This is a data-partition using the HPFS file-system.

Note that this is on a logical partition and can only be booted by OS/2-BMGR or AiR-BOOT.

We have it here to demonstrate AiR-BOOT behavior to initially show all partitions.

J: Partition

This is a data-partition using the JFS file-system.

Note that this is on a logical partition and can only be booted by OS/2-BMGR or AiR-BOOT.

We have it here to demonstrate AiR-BOOT behavior to initially show all partitions.

L: Partitions

These are spanned partitions using OS/2 Logical Volume Management to provide a spanned volume.

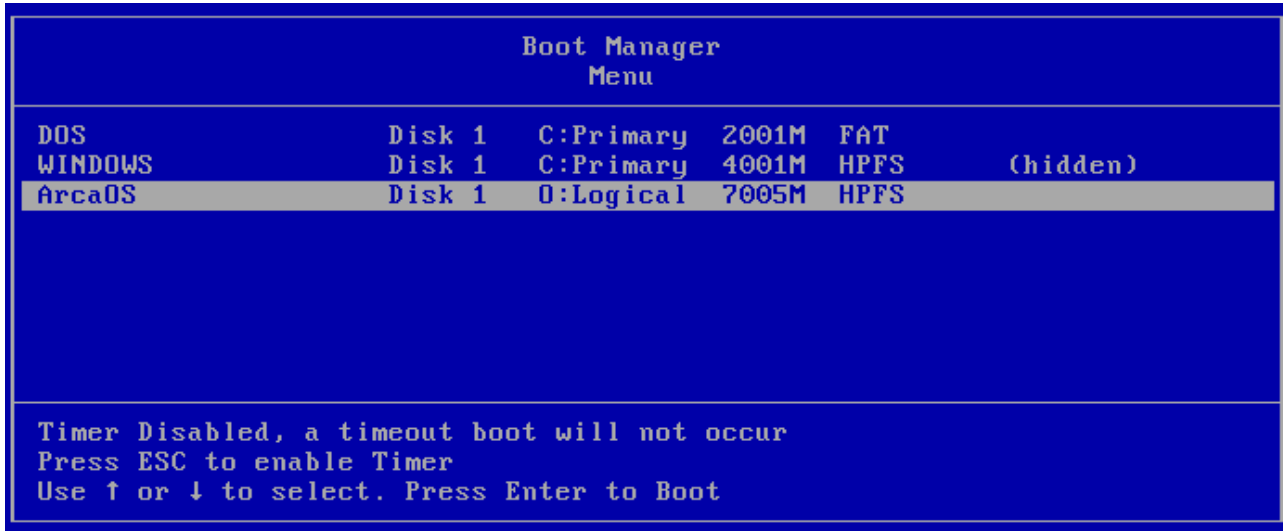
Such partitions are not bootable and we have them here to demonstrate AiR-BOOT behavior to initially hide them.

O: Partition "ArcaOS"

This partition contains the ArcaOS installation on the JFS file-system.

Note that it is a logical partition and thus needs OS/2-BMGR or AiR-BOOT to boot from it.

Initial OS/2-BMGR Menu



This is the menu that OS/2-BMGR presents with our sample configuration. We are going to replace OS/2-BMGR with AiR-BOOT and when that is done you can compare the AiR-BOOT menu with this one.

Installing AiR-BOOT

Now we are going to install AiR-BOOT to take over the boot functionality from OS/2 Boot Manager.

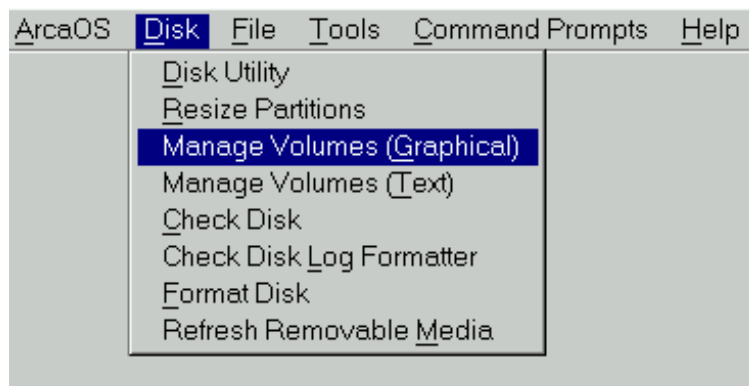
Since AiR-BOOT is specifically enhanced to integrate with OS/2 and ArcaOS, the first method that will be discussed is the GUI method of installation using the Graphical Volume Manager on the ArcaOS Installation Medium. However, AiR-BOOT was designed to be operating-system agnostic, meaning it can be installed from other operating systems too by using the command-line programs.

Installing using the ArcaOS Installation Medium

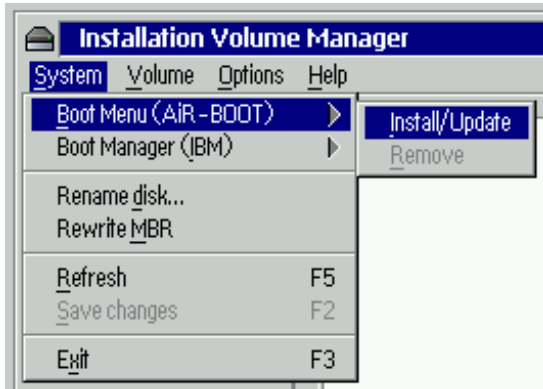
Boot from the ArcaOS Installation Medium and choose to boot from CD.

At the welcome screen, choose System Management.

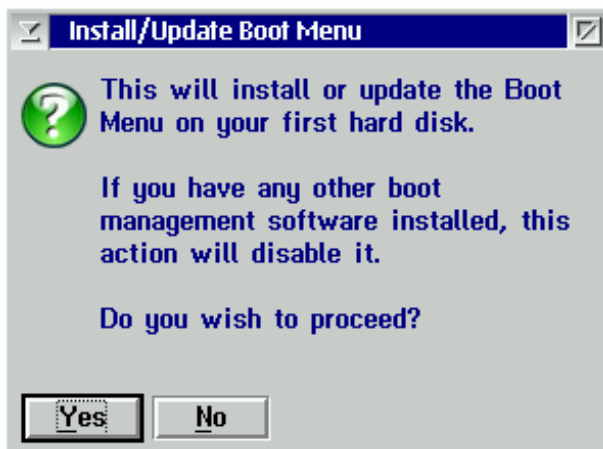
Once in System Management, start the Graphical Volume Manager.



From the Graphical Volume Manager, choose the following from the menu:



A confirmation dialog will appear:



Press the Yes button to install AiR-BOOT.

Now you can shutdown the system and start from the hard-disk, which will present the AiR-BOOT Menu.

Some things to be aware of

Atomic operation

Note that, contrary to other operations in MiniLVM, the installation of AiR-BOOT is atomic, meaning it takes place immediately and does not require you to save the changes. You can easily revert by choosing Remove, and make sure OS/2-BMGR or some other primary partition is marked Startable. In this case you do have to save the changes because the startable state has been changed.

Disabled Remove Menu Entry

Also note that in the above example, the Remove entry in the menu is grayed-out, and not selectable. This is because AiR-BOOT was not installed in the example above. If there was an AiR-BOOT installed then this entry would have been selectable, and choosing it would have removed AiR-BOOT. Please refer to the section Removing AiR-BOOT for more details.

Install/Update Menu Entry

If AiR-BOOT was already installed, the Install/Update action would upgrade AiR-BOOT to the version that is present on the ArcaOS Boot-CD, or, if you are using MiniLVM from an installed system, to the version installed with that. Upgrading only takes place if that version is of a higher level than the one already installed, otherwise no action is taken. See the section for Upgrading AiR-BOOT for more details.

Possible OS/2-BMGR Menu Entry

Also note that when OS/2-BMGR is installed, the above menu will look different. It will also show a menu-entry named "Boot Manager (IBM)", which would normally manage OS/2-BMGR, if AiR-BOOT were not installed. With AiR-BOOT installed, this entry is still present to indicate the presence of OS/2-BMGR, but you will not be able to manage it because AiR-BOOT is now the boot-manager in control.

Installing using the OS/2 command-line

You can also install AiR-BOOT from the command-line. This offers a method with more control over the AiR-BOOT installation. One thing is that you can downgrade to a previous version or that you can rewrite a default configuration.

Note however that the use of the command-line tool is considered *advanced usage* and you are expected to know what you are doing. Another thing to be aware of is that v1.07 contains a bug with regard to saving the full configuration, so if you do a code only downgrade from v1.1.0 or higher this might produce unexpected results. Code downgrading to the v1.06 version is not supported at all at this time since it's configuration is of lower level. The only way to downgrade is to also rewrite the configuration corresponding to the code-version.

The AIRBOOT2.EXE installer

This is the OS/2 command-line installer, `airboot2.exe`, and it needs the `airboot.bin` file to be present in the same directory. Running it without any option will check if AiR-BOOT is currently installed. It does this by checking the Master Boot Record for the AiR-BOOT signature. If AiR-BOOT is not installed it will offer to install it. If there is already an AiR-BOOT installed, it will offer to upgrade it; no action will be taken if the present version is of a higher version. If it finds a damaged AiR-BOOT, like the signature is present in the MBR but the code or configuration is missing or corrupted, it will offer to repair it. A damaged or disabled AiR-BOOT configuration can easily be repaired by using this command-line program and 9-out-of-10 times your configuration will still be intact.

Reinstalling AiR-BOOT in case of a corrupted boot-environment is almost always the best thing to try first.

The '/forcecode' flag on the installer

This flag forces the installer to rewrite the AiR-BOOT code.

It is of value in two situations:

- ◆ You want to downgrade to a previous version
Take into account the previous section about downgrading
- ◆ You are testing a beta
In this case, the code version of the beta is the same as that of the final release that will follow it, so you need to force the code upgrade otherwise no action will be taken

The '/forceconfig' flag on the installer

The AiR-BOOT configuration has it's own version and it is divided in two parts:

A part that has the initial settings, like whether to time-out or not
This could be referred to as the 'semi-static' or 'core' configuration

A part that is dynamic, like the partitions that were found on your disks at the last boot
This changes every time you add or remove partitions

AiR-BOOT User Manual

Version 1.1.4

Below are some screen-shots of running the OS/2 installer from the command-line:

```
AiR-BOOT Installer v1.1.4 for OS/2
- (c) Copyright 1998-2012 by Martin Kiewitz.

-> ...Please wait... <-
- Loading bootcode from file...ok
- Loading MBR from harddisc...ok

-> ...Current Status... <-
- AiR-BOOT is not installed
- Configuration is intact (v1.1.0)

-> ...!ATTENTION!... <-

-> ...Please press... <-
<A> - Add AiR-BOOT 'v1.1.4/english' on current system
<D> - Delete AiR-BOOT from current system
<Q> - Quit without any change
```

As you can see, it states in -> ...Current Status... <- that AiR-BOOT is not installed yet and offers the option A to add it. After pressing A, the following will be shown:

```
Add/Repair/Update AiR-BOOT in progress...
CB74E5BC
  ■ Writing AiR-BOOT code...ok

Your copy of AiR-BOOT is now fully functional.
Please hit ESC to exit AiR-BOOT installer or ENTER to reboot your system...
```

Of course, it is best to press ESC and do a proper shutdown and reboot of the system.

Installing AiR-BOOT from other Operating Systems

There are no GUI front-ends (like MiniLVM) available for other operating-systems to install AiR-BOOT, so this must be done from the command-line. Like the command-line installer for OS/2, there are the `airbootd.exe` installer for DOS and the `airbootw.exe` installer for Windows. A Linux command-line version, `airbootl.elf`, is in the works and may appear in future releases. Also note that the command-line installers for DOS and Windows require certain environments to be run successfully. For DOS this means *no virtualization*, like a DOS Box and for Windows NT/2K/XP/Vista/7 it means running them in a Command Prompt that was started as an *Administrator*.

AiR-BOOT User Manual

Version 1.1.4

Installing AiR-BOOT from DOS

The AiR-BOOT installer needs to physically access the hard-disk to install itself. This means that any "emulated" or "virtualized" DOS-environment may pose limitations on accessing real hardware and may interfere with the installation. Even worse, it may cause damage to your disk !

Therefore, it is mandatory that you use a "Real DOS", like one you boot from floppy or hard-disk, to run the `airbootd.exe` installer . The recommended way is to create a "FreeDOS" bootable diskette and copy the `airbootd.exe` installer and `airboot.bin` files to that and then boot this disk and run the installer. You may also use PC-DOS7 or MS-DOS5, but don't use versions lower than 5 for safety reasons. The flags that can be used are the same as for the `airboot2.exe` installer: `/forcecode` and `/forceconfig`.

Installing AiR-BOOT from Windows

There are roughly two "families" of Windows:

- ◆ The Windows 1.x/2.x/3.x/9x/ME versions
These versions actually run on DOS and thus use the DOS kernel. Use the `airbootd.exe` installer in a DOS Window (1/2/3.x) or DOS-MODE (9x/ME) in these cases. Better yet, boot from a plain FreeDOS (diskette) for the installation and forget about installing from Windows.
- ◆ The Windows NT/2000/XP/Vista and 7 versions
Believe it or not, these versions are remotely based on OS/2 and thus use their own kernel. Use the `airbootw.exe` installer from a Command Prompt opened as Administrator. Find info on the Windows "RunAs" command for more info if needed. Vista and 7 implement some brain-dead 'solutions' to the admin/no-admin paradigm in the form of the UAC, (User Access Control). If this piece of wasted bits pops-up try to convince it that you are in control.

A note on installing from Windows

The DOS based Windows versions put many "tentacles" in the vulnerable DOS kernel code, especially for accessing mass storage devices. Since the TRACK0 area is an area of the disk that is normally out-of-scope for regular file-access, unpredictable things can happen. The NT-and-up versions are more reliable but the installer has not been tested as extensively on these versions as it has been on OS/2.

Installing AiR-BOOT from Linux

A Linux versions of the installer is in the works but at the time of this writing not ready yet. However, it seems a good idea to provide some info in advance. Most Linux versions come with the GRUB boot-manager. There are two versions of GRUB: the legacy version (0.9x) and the new (1.9x) version. The new GRUB is a complete rewrite and differs considerably from the legacy version with regard internal operation and configuration / maintenance. In both cases however, a Linux installation will probably have installed the GRUB loader in the MBR, so installing AiR-BOOT would disable GRUB and make your Linux systems unbootable. AiR-BOOT cannot directly boot Linux but it can chain-load GRUB. So to boot Linux from AiR-BOOT the GRUB loader has to be relocated to a partition. See further down in this manual on how to do that. If you are planning to install a new Linux system and want to use AiR-BOOT too then place grub into the partition you install Linux into and not in the MBR.

Using AiR-BOOT in other languages

The default language for AiR-BOOT is English. However, AiR-BOOT also supports:

- ◆ German
- ◆ French
- ◆ Italian
- ◆ Dutch
- ◆ Russian
- ◆ Swedish.

To use an alternate language, copy the corresponding file, called `airbt-?.bin` to `airboot.bin` before running the installer. The ?? is the 2-letter language identifier.

AiR-BOOT User Manual

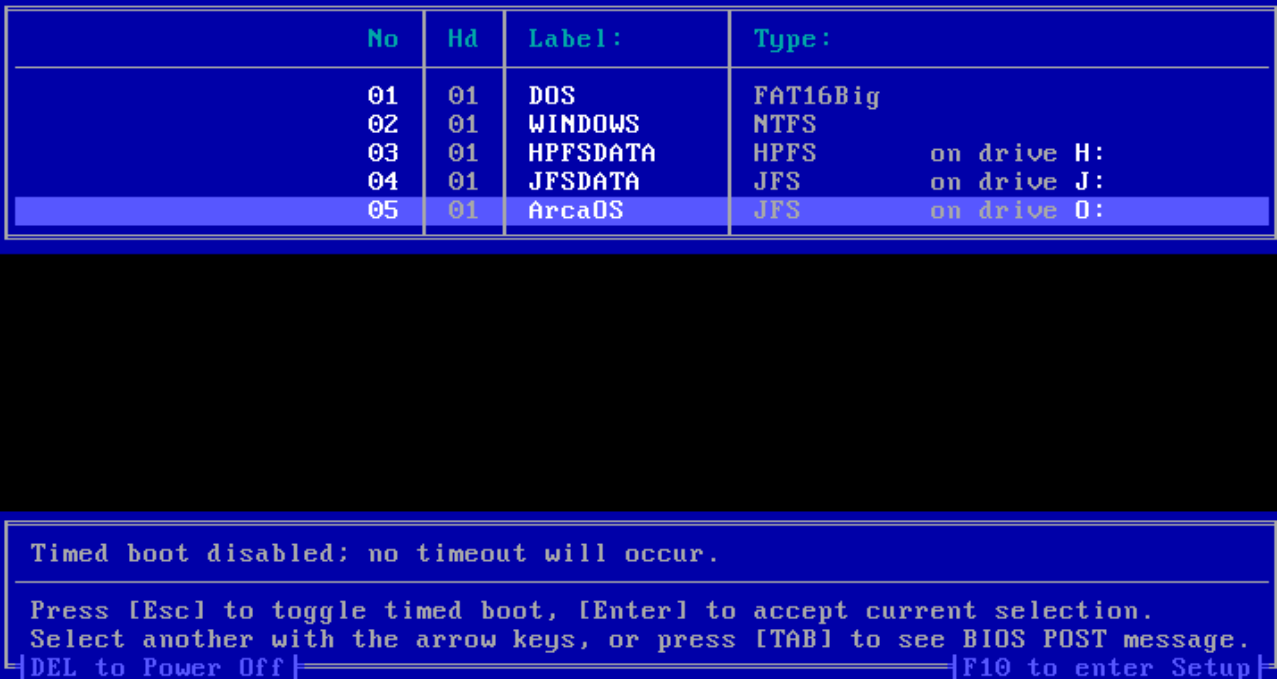
Version 1.1.4

In general

Refrain from using Windows to install AiR-BOOT. (Better yet, refrain from using it at all)
When installing Linux, put GRUB in a logical partition so it can be chain-loaded by AiR-BOOT.

Now that we have managed to install AiR-BOOT, let's see how it works and what we can do with it...

Initial AiR-BOOT Menu



No	Hd	Label :	Type :
01	01	DOS	FAT16Big
02	01	WINDOWS	NTFS
03	01	HPFSDATA	HPFS on drive H:
04	01	JFSDATA	JFS on drive J:
05	01	ArcaOS	JFS on drive O:

Timed boot disabled; no timeout will occur.

Press [Esc] to toggle timed boot, [Enter] to accept current selection.
Select another with the arrow keys, or press [TAB] to see BIOS POST message.
[DEL] to Power Off [F10] to enter Setup

As you can see, the AiR-BOOT Menu pretty much resembles the OS/2-BMGR Menu we left behind.

But why are the HPFSDATA and JFSDATA partitions in the menu ?

The short answer is that AiR-BOOT is an on-the-fly boot-manager and has no real means of determining whether a partition is bootable or not. Thus, when AiR-BOOT is installed for the very first time, it assumes all partitions are bootable and shows them in the menu. The only exceptions are type 0x35 (LVM spanned) partitions, which are not bootable anyway, and a 0x0a (OS/2-BMGR) partition if present. While a 0x0a partition can be booted, chain-loading OS/2-BMGR has to be explicitly enabled by the user.

It is up to the user to specify which partitions contain bootable systems and which do not. This is done by configuring AiR-BOOT using its built-in SETUP. This is a one-time action after initial installation and needs only to be repeated after data partitions are added.

The following section explains more about operating AiR-BOOT and how to hide non-bootable partitions from the menu.

Operating AiR-BOOT

The AiR-BOOT Menu does not only resemble OS/2-BMGR, it can also be navigated using the same keys like UP and DOWN to move the cursor, ESC to enable/disable the timer and ENTER to boot the selected system. Additionally there is the F10-key, which is used to enter the built-in SETUP of AiR-BOOT.

Explanation of the screen layout

The No column

This column shows the **partition number / index**. This index is a logical index and does not relate to the order by which partitions are stored on the disk(s). Manually adding or removing a partition from the AiR-BOOT menu will renumber the list.

The Hd column

This column identifies the **hard-disk** that the partition is on. For convenience the partition list is ordered by this column, so all partitions on the first (boot) disk are displayed first, then all other partitions on other disks. Note however, that AiR-BOOT disk enumeration depends on the BIOS and that a running OS/2 system may present disks in a different order, depending on loaded drivers, filters and disk types.

The Label column

This column shows the **Label** of the system.

AiR-BOOT takes this label from OS/2 LVM-info if present, from a BPB in case it supports one, or it displays the type of the file-system when no other label-source can be found. While AiR-BOOT is capable of editing labels, you should familiarize yourself with how they are handled because of the different origins they might be taken from. The safest way is to label your partitions using MiniLVM because the presence of LVM information takes precedence over all other label-sources in AiR-BOOT. Due to TRACK0 space restrictions labels are always truncated to 11 characters in this version of AiR-BOOT. So if you used MiniLVM to give a longer name, only the first 11 characters will be shown here. This restriction might be lifted in future versions.

The Type column

This column shows the **type of the system**. It is not directly a file-system identifier (but mostly is), but rather more a combination of file-system and/or partition-type. A common misconception is that the type of a partition indicates a particular file-system. This is not the case since for instance, type 0x07 indicates an *installable file-system*, which can be HPFS, JFS, NTFS or even exFAT. Likewise a Linux partition of type 0x83 can host a multitude of file-systems, like EXT2/3/4, JFS, XFS, REISER3/4, BTRFS and whatnot. And in the case of OS/2 Boot Manager (type 0x0a) there is no file-system to speak of, while in an opposite situation, that of many vendor's "recovery partitions", some custom type may be used that actually holds a FAT or otherwise "well known" file-system. As a rule of thumb: A partition type *does not* indicate a particular file-system !

The bottom screen area

Besides displaying the Timer, this area displays the most important keys for controlling AiR-BOOT. Esc to toggle the timer, ENTER to boot the selected partition, TAB to switch between the menu and post-boot info and finally the F10 key to enter AiR-BOOT SETUP.

Navigating AiR-BOOT

There is not much to say here other than that the bottom screen says it all. Arrow UP/DOWN to get the cursor on the system you want to boot and press ENTER to boot it. Just like OS/2 Boot Manager.

However, navigating the AiR-BOOT SETUP has a specific behavior because some changes are 'atomic' and other changes require 'saving the configuration'. These will be dealt with in the corresponding section.

Why almost all partitions are initially listed, even non-bootable ones

At first this may seem strange behavior to you, but take into consideration that AiR-BOOT works on-the-fly every time it is started. At every boot, it rediscovers your disks and partitions. The only partitions that are hidden by default are OS/2 Boot Manager and spanned LVM partitions. Below is a more thorough explanation of this behavior.

Removing instead of adding

In OS/2-BMGR you have to manually add any partitions you want to boot to the OS/2-BM menu, using tools like MiniLVM or ClassicLVM. AiR-BOOT needs the reverse: you have to remove any partitions from the menu you don't want. This is because it enumerates partitions on-the-fly each time it is started, thus every time you boot your system. It always shows all partitions after a fresh installation, whether they are bootable or not. (*Except OS/2-BMGR or Spanned LVM-Volumes*) So it's the responsibility of the user to remove any unwanted partitions from the AiR-BOOT menu. This is a one-time action and is only needed after a fresh installation of AiR-BOOT or when new data partitions are created. When you decide to add a partition to OS/2-BMGR, you say it is bootable and OS/2-BMGR will happily try to boot it. Even if it's a non-bootable partition. AiR-BOOT does the reverse: it adds what it finds and requires you to say it's not bootable. This "inverse logic" actually makes much more sense since any bootable system you add will be found without requiring your intervention and any non-bootable systems are removed with the same ease as adding bootable ones to OS/2-BMGR.

Why AiR-BOOT cannot safely determine whether a partition is bootable or not

When AiR-BOOT starts, right after getting control from the BIOS, it enumerates all partitions on all disks. Since OS/2 can also be booted from logical partitions, to distinguish such a bootable partition from a data partition would require further investigation with the need to access on the file-system level. And OS/2 is not the only system that can be booted from a logical partition, so it would require much more interrogation to determine whether such a logical partition is truly bootable or not. This also goes for primary partitions, since these could be data partitions too. Since AiR-BOOT cannot access partitions on the file-system level, it has no true means to determine if a partition is bootable or not.

The meaninglessness of the 'bootable' or 'active' flag for AiR-BOOT

The 'bootable' or 'active' flag is meaningless to AiR-BOOT since AiR-BOOT is an interactive boot-manager. The bootable flag is used by simple MBR-code that does not interact with the user and just boots the partition that has this flag set. That's also why, when uninstalling AiR-BOOT, you need to take care that a valid primary partition is marked bootable, otherwise you could end up with a non-bootable system.

Removing partitions from the AiR-BOOT menu is the user's responsibility

So, instead of AiR-BOOT making uneducated guesses about the bootability of the partitions it finds, it's up to the user to filter which partitions should be displayed and which should be hidden from the AiR-BOOT menu. This is a one time configuration effort that needs to be performed after a fresh install of AiR-BOOT or when non-bootable partitions are added. Partitions that are physically removed will just not be found when AiR-BOOT starts and will thus automatically disappear from the menu. The only partitions that AiR-BOOT always hides after a fresh install are the OS/2-BMGR partition and partitions that are part of an LVM Spanned Volume, type 0x35, which are not bootable anyway.

AiR-BOOT User Manual

Version 1.1.4

Removing partitions from the AiR-BOOT menu is completely unrelated to hiding partitions by changing their type

Note that the removal of partitions from the AiR-BOOT menu has nothing to do with the hiding of partitions by altering their partition type to a so called hidden type ! The hiding mentioned above is a feature of AiR-BOOT and only controls the visibility of a partition in the AiR-BOOT menu, conceptually equal to removing or adding a boot-entry from or to the OS/2-BMGR menu. Removing a partition from the AiR-BOOT menu does not alter the partition or the partition tables in any way.

Now that we have covered these things, let's see how we can make this initial AiR-BOOT boot-menu look more like the OS/2 Boot Manager boot-menu we left behind.

Hiding the non-bootable partitions

Enter the AiR-BOOT Setup by pressing F10, and choose PARTITION SETUP which will show the following screen:

No Hd	Label	Flags	Type	No Hd	Label	Flags	Type
01/01	DOS	BUHLP	FAT16Big	02/01	WINDOWS	BUHLP	NTFS
03/01	OS2 BootMgr	BUHLP	OS2-BMGR	04/01	HPFSDATA	BUHLP	HPFS
05/01	JFSDATA	BUHLP	JFS	06/01	LUMDATA	BUHLP	LUM-Data
07/01	LUMDATA	BUHLP	LUM-Data	08/01	ArcaOS	BUHLP	JFS

↑ ↓ → ← : Choose partition
Enter : Edit label

F1 : Flags (press key to toggle)
Esc : Return to main-menu

Note the bold green B-flags on the HPFSDATA and JFSDATA partitions which causes them to be shown in the menu.

Use the UP/DOWN and LEFT/RIGHT keys to move the cursor to the HPFSDATA partition and press the b-key. This will turn-off the B-flag for that partition. Now do the same for the JFSDATA partition. After removal of the B-flags, the screen looks like this:

No Hd	Label	Flags	Type	No Hd	Label	Flags	Type
01/01	DOS	BUHLP	FAT16Big	02/01	WINDOWS	BUHLP	NTFS
03/01	OS2 BootMgr	BUHLP	OS2-BMGR	04/01	HPFSDATA	BUHLP	HPFS
05/01	JFSDATA	BUHLP	JFS	06/01	LUMDATA	BUHLP	LUM-Data
07/01	LUMDATA	BUHLP	LUM-Data	08/01	ArcaOS	BUHLP	JFS

↑ ↓ → ← : Choose partition
Enter : Edit label

F1 : Flags (press key to toggle)
Esc : Return to main-menu

Now press ESC to return to the Main Setup Menu and then F10 to save the configuration.

The AiR-BOOT Menu is displayed again and now the data-partitions are not shown:

No	Hd	Label:	Type:
01	01	DOS	FAT16Big
02	01	WINDOWS	NTFS
03	01	ArcaOS	JFS on drive 0:

Timed boot disabled; no timeout will occur.

Press [Esc] to toggle timed boot, [Enter] to accept current selection.
Select another with the arrow keys, or press [TAB] to see BIOS POST message.
[DEL to Power Off] [F10 to enter Setup]

Note that the B-flag is a bit of a misnomer, it has nothing to do with magically making a partition bootable, nor does it have any relation to the boot-flag known from partition-table entries. Its meaning is: this partition is bootable and it is the user who specifies that.

Does this not look surprisingly much like the OS/2 Boot Manager screen we left behind ?

Configuring AiR-BOOT

Unlike OS/2-BMGR, AiR-BOOT does not require any external tools to manage it's configuration. The next sections will cover the configuration of AiR-BOOT.

Entering and Leaving the Sub Sections

Below are the keys used to enter and leave the sub-menu's in the AiR-BOOT Setup.

Entering the SETUP (F10)

To enter the AiR-BOOT SETUP, press F10.

Having the cursor on an entry and pressing F1 will show a pop-up with information about what actions can be performed.

Entering a Setup Section (ENTER)

Place the cursor on the entry you want and then press ENTER.

The corresponding setup-screen for the section will then appear.

Leaving a Setup Section (ESC)

Press the ESC key to return to the Main Setup Menu.

Now you can either save your changes by pressing F10 or discard them by pressing ESC.

AiR-BOOT User Manual

Version 1.1.4

The AiR-BOOT SETUP Menu

PARTITION SETUP	DEFINE MASTER PASSWORD
BASIC OPTIONS	DEFINE BOOT PASSWORD
ADVANCED OPTIONS	SAVE & EXIT SETUP
EXTENDED BOOT OPTIONS	EXIT WITHOUT SAVING
↑ ↓ → ← : Choose Action	F10 : Save&Exit Setup
Enter : Select Action	Esc : Quit Setup

PARTITION SETUP

No Hd	Label	Flags	Type	No Hd	Label	Flags	Type
01/01	DOS	BUHLP	FAT16Big	02/01	WINDOWS	BUHLP	NTFS
03/01	OS2 BootMgr	BUHLP	OS2-BMGR	04/01	HPFSDATA	BUHLP	HPFS
05/01	JFSDATA	BUHLP	JFS	06/01	LUMDATA	BUHLP	LUM-Data
07/01	LUMDATA	BUHLP	LUM-Data	08/01	ArcaOS	BUHLP	JFS
↑ ↓ → ← : Choose partition				F1 : Flags (press key to toggle)			
Enter : Edit label				Esc : Return to main-menu			

The first thing you are likely interested in is hiding the data-partitions from the menu. So we press ENTER and a screen similar to this will appear:

Note that the SETUP menu is actually composed of two halves, each displaying about half of all partitions present. Use the arrow-keys to navigate over the entries, press ENTER to customize the label.

The BVHLP Flags

These are shortly explained in the HELP pop-up you can activate by pressing F1.

Since AiR-BOOT comes in several languages these letters are language dependent, so if you use a non-English version of AiR-BOOT please substitute them in your mind when reading the following paragraphs.

Each letter in the Flags field is actually a 'hot-key' and pressing it will toggle the corresponding functionality or pop-up a secondary menu where advanced settings can be made. Both the 'H' and the 'L' activate such an advanced menu. When a feature is active, it's letter will display highlighted, otherwise it's dimmed.

The Bootable feature -- show or hide a partition from the main boot-menu

This is the thing you were probably looking for when you installed AiR-BOOT for the very first time and wondered how to convince it to leave out your data partitions. Pressing it toggles the visibility of the partition in the main boot-menu. In the default configuration this option is set for all partitions except the OS/2-BMGR partition if present and LVM spanned data partitions.

Depending on your point-of-view, this term "bootable" can be a bit misleading. It does not make a data partition bootable by magically installing partition loader code on it, nor does it remove loader code from a partition that is bootable. The term Bootable refers to you

AiR-BOOT User Manual

Version 1.1.4

wanting to boot that partition and thus having it displayed in the main boot-menu. Setting the B on a non-bootable partition will show it in the boot-menu, but booting it will of course fail. That is, unless some cosmic radiation has rearranged the magnetic particles on your disk to exactly match the pattern for a partition loader for the x86 cpu architecture...

The Virus Protect feature -- enable / disable virus checking

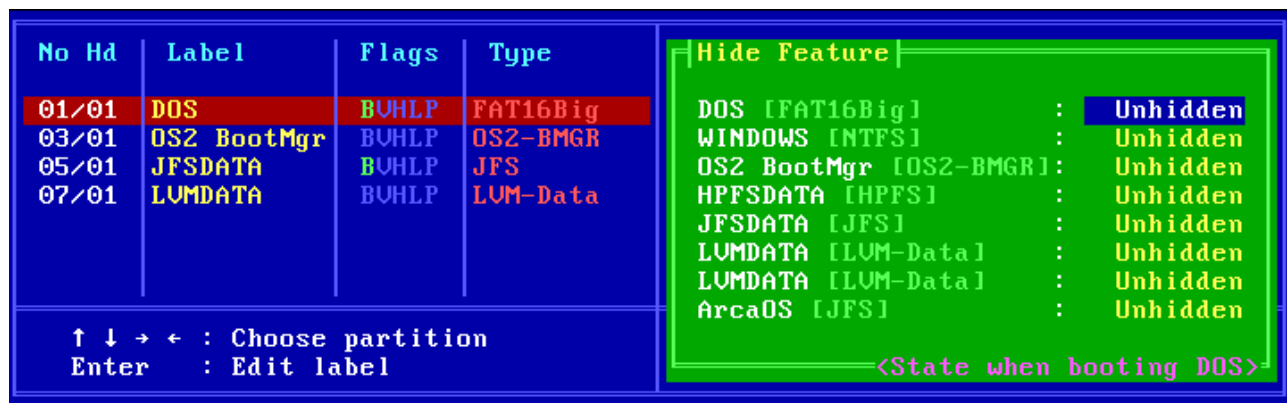
The loader-code has the ability to protect itself from boot-sector viruses.

It does this by checking whether some interrupt pointers point to ROM or RAM areas. If such a pointer points to a RAM area, the loader-code assumes some malicious program has intercepted the vector because normally control is directly transferred from the BIOS to AiR-BOOT. In such a case AiR-BOOT will refuse to operate any further and a hard reset of the system is required. With today's advanced systems however, it's possible that this check intervenes with system operation. If that's the case you can turn this option off by pressing V and save the new configuration.

The Hide Partition feature -- hide one or more partitions

This is an enhanced feature that shows a popup-menu where you can set which partitions need to be hidden when the corresponding entry is booted. So this is a per-system setting. Some legacy systems, like DOS and also the DOS based Windows systems like Win3x, Win95/98/ME, use partition-loader code that gets confused when multiple primary partitions formatted with FAT are visible. It can happen that you boot the third primary containing PC-DOS7 but that the PBR-code of PC-DOS7 actually boots your DOS 3.3 from the first primary. Using this feature you can mark the partitions you want hidden when booting the corresponding entry.

The advanced pop-up looks like this:



At the bottom of the green advanced pop-up you can see to which partition the hide feature will be applied.

It corresponds to the entry highlighted in the main blue menu, in this case PC-DOS7.

Pressing the SPACEBAR will toggle the Unhidden/Hidden status in the green advanced pop-up, UP/DOWN will navigate.

Also, pressing F1 will show a compact help pop-up.

The Drive Letter feature -- Forcing an LVM drive-letter

This option only effects OS/2 LVM drive-letter assignments.

It does not make DOS boot from D: or Windows from W:

It's main function is to force a drive-letter on an OS/2 bootable partition and it's main usage is to have multiple OS/2 installations using the same boot-drive-letter. It will not magically make OS/2 boot from another drive-letter then it was installed to !

Pressing L will show a pop-up and pressing a letter will assign that letter to the LVM-information for that partition.

Pressing BACKSPACE will clear the assignment.

AiR-BOOT User Manual

Version 1.1.4

BASIC OPTIONS

Here you can set some basic options and the screen will resemble something like this:

Default Selection : ArcaOS	Include Floppy Drive : Enabled
Timed Boot : Enabled	Show LUM Drive Letters : Enabled
Timed Boot Delay (sec) : 15	MBR Protection : Disabled
Timed Key Handling : Do Nothing	-> Ignore MBR Writes : Disabled
Boot From Last If Timed : Enabled	Make Sounds : Disabled
Remember Last Boot : Enabled	
Remember Timed Boot : Disabled	
↑ ↓ → ← : Choose Item	F1 : Show help for Item
PgUp/Dn : Change Item	Esc : Return to main-menu

Default Selection

This is where you can specify which system you want to boot by default. Press the SPACEBAR to cycle through all entries that have the B-flag set. Note that this option has no effect when "Remember Last Boot" is enabled.

Timed Boot

This option enables / disables the timer. A default AiR-BOOT installation from scratch has the timer enabled, so here is where you can turn it off. Simply pressing ESC in the main AiR-BOOT menu toggles the timer activity for the current boot only.

Timed Boot Delay

Speaks for itself; it is the boot-delay in seconds when a timed boot is enabled. Use PgUp/PgDn to decrease/increase. (yes, it's a bit counter intuitive)

Timed Key Handling

Here you can set what you want AiR-BOOT to do when a key is pressed while counting down to boot. The options are:

- ◆ Do Nothing
- ◆ Reset Time
- ◆ Stop timer

Press the SPACEBAR to cycle to the above options.

Boot From Last If Timed

When this option is enabled, AiR-BOOT will boot the entry that was last booted when using timed-boot.

Remember Last Boot

When this option is enabled, AiR-BOOT will remember the last non-timed boot and the "Default Selection" has no effect.

Remember Timed Boot

When this option is enabled, AiR-BOOT will remember the last timed boot.

Include Floppy Drive

Enabling this option will show booting from floppy in the AiR-BOOT Main Menu.

AiR-BOOT User Manual

Version 1.1.4

Show LVM Drive Letters

Enabling this option will show the corresponding drive-letter for each OS/2 bootable partition.
It does not show drive-letters for DOS or Windows systems since their assignments are handled differently from OS/2 LVM.

MBR Protection

This is a left-over from the era where DOS viruses could infect the MBR.
With the enhancements made to AiR-BOOT this feature is currently broken and will probably be removed or changed in future releases.
Best is to leave it off.

Ignore MBR Writes

This feature is also non-functional and should not be changed.

Make Sounds

When this option is enabled, AiR-BOOT will treat you with some audible feedback.

ADVANCED OPTIONS

Here you can set a number of advanced options and the screen will resemble something like this:

Boot Menu : Enabled	Get Floppy Name : Disabled
Partitions Autodetect : Enabled	-> Get Every 2 Seconds : Disabled
-> Security Options <-	Virus Detection : Disabled
Passworded Setup : Disabled	Stealth Detection : Disabled
Passworded System : Disabled	UIBR Detection : Disabled
Passworded Change Boot : Disabled	Continue BIOS Boot Seq : Disabled
↑ ↓ → ← : Choose Item	F1 : Show help for Item
PgUp/Dn : Change Item	Esc : Return to main-menu

Boot Menu

Here you have the following options:

Enabled

This is the default and it shows the standard AiR-BOOT Menu.

Detailed

Choosing this option will show the sizes of the partitions in the Hd-column.
However, at the last moment I noticed there is a border-drawing flaw such that the left and right border are not drawn correctly.

AiR-BOOT User Manual

Version 1.1.4

Disabled

This will keep AiR-BOOT from displaying the Main Menu when started. (As if it was not installed at all)

It will directly boot the Default Selection you have set. If you are a die-hard fan of the OS/2 Boot Manager, you can enable it by setting the B-flag on it's partition, setting it as the Default Selection and then disable the AiR-BOOT Menu. AiR-BOOT will then transparently boot into OS/2-BMGR giving you that "good old feeling" :-)

When the AiR-BOOT Menu is disabled, there is no user interaction so F10 is not available to enter AiR-BOOT Setup. In that case, hold down on of the Alt-keys when AiR-BOOT is starting and you will be thrown into AiR-BOOT Setup.

Partitions Autodetect

As you know by now, AiR-BOOT scans for partitions every time it is started.

New partitions are added to the menu and removed partitions are deleted from it.

Setting this option to disabled makes AiR-BOOT do no scans and use the previous configuration.

Thus, when disabled and you made major modifications to you hard-disk layout, things might go very weird !

Security Options

Here you can specify at what places AiR-BOOT should ask for a password before proceeding.

This is handy on machines that are part of a more controlled environment.

The MASTER and BOOT passwords can be set using the main setup menu. (*as will be shown later*)

Get Floppy Name

This feature is not working currently.

The idea is to show the floppy-label in the menu when booting from floppy is enabled.

Best is to keep it disabled.

Virus Detection

This option globally enables or disables virus detection for the two entries below it.

Master Boot Sector and Partition Boot Sector viruses stem from the DOS era and are not very common anymore.

Many of them have been rewritten to Word Macro's or ActiveX, where they found a much more comfortable home :-)

Also, with the enhancements made in post v1.06 versions, the virus detection has not been thoroughly tested.

Another thing is that enabling it may interfere with protected mode operating systems like OS/2, so it's best to keep the virus detection disabled.

Stealth Detection

When this option is enabled, an attempt will be made to discover stealth viruses.

Again, this has not been tested with the enhancements made to AiR-BOOT, so better leave it disabled.

VIBR Detection

This checks to see if a Partition Boot Record has been changed since the last time and possibly could contain a virus.

For OS/2 however, the PBR will be changed when using the enhanced drive-letter feature.

While such a change will update the CRC of the PBR in the AiR-BOOT Configuration, it's also better to leave this one disabled.

Continue BIOS Boot Seq

This option re-transfers control to the BIOS for a particular device.

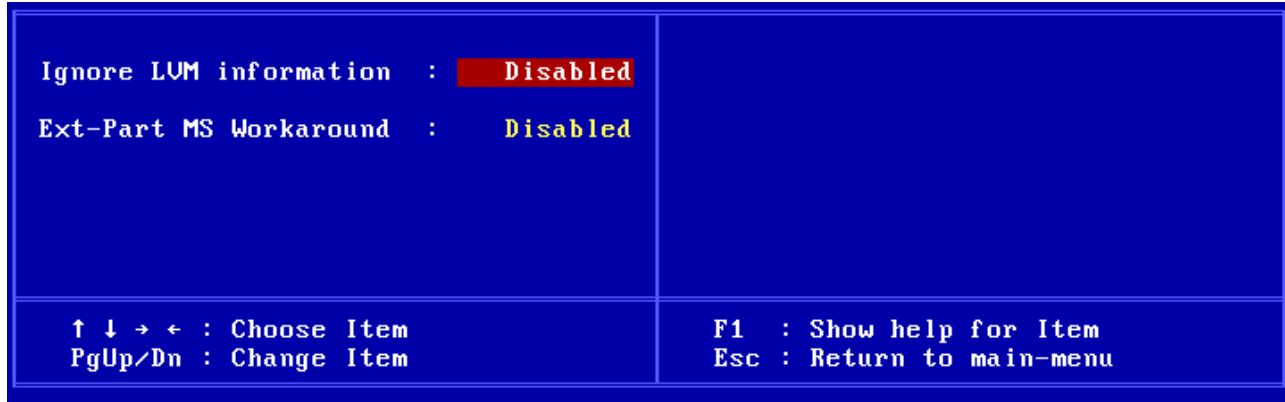
The choices are:

- ◆ Disabled
- ◆ CD-ROM
- ◆ Network
- ◆ ZIP/LS120

This is highly BIOS dependent and most options here do not work on modern BIOSses.

EXTENDED BOOT OPTIONS

Here you can set some (less useful) options for booting and the screen will resemble something like this:



Ignore LVM information

When this option is enabled AiR-BOOT will ignore LVM information if present.

If you use OS/2, it is highly recommended to leave this option disabled.

Ext-Part MS Workaround

When this option is enabled, the partition identifier of the extended container will be changed to comply with "MS Inventions".

Just leave it disabled if non of your doze-systems complain.

A normal extended partition container has an identifier of 0x05.

However, Microsoft found it necessary to also introduce the 0x0f identifier to indicate LBA usage.

As usual they don't consider the possibility of other operating systems being used and even less that they could be booted from a logical partition like OS/2.

DEFINE MASTER PASSWORD

Here you can define the master password.

You will be presented with a pop-up to enter it.

Entering nothing will clear the password.

(After giving the correct one if one is set)

DEFINE BOOT PASSWORD

Same as for MASTER PASSWORD.

SAVE & EXIT SETUP

Save your setup changes to the AiR-BOOT Configuration which is stored in track zero.

EXIT WITHOUT SAVING

Do not save your changes.

Not however that some changes are 'atomic', like forcing a drive-letter to an LVM partition and even if you exit without saving, the change will be recorded because the configuration is saved when a system is booted.

This behavior will be improved upon in future releases.

Upgrading AiR-BOOT

This version (v1.1.4) can upgrade AiR-BOOT from v1.06, v1.07, v1.1.0 and v1.1.2.

Most if not all configuration settings of the previous version will be taken over, however it's a good idea to check your configuration after an upgrade.

Upgrading from pre-v1.06 versions

Upgrading from v0.9x version should work but it has not been tested.

Upgrading from older versions than v0.9x is not supported, so you'll have to remove the old version and reinstall the new version after which you need to reconfigure AiR-BOOT.

Notice about upgrading from v1.07

Due to the bugs in the v1.07 release, the hidden partitions are cleared, so you have to reconfigure these if you used this feature.

Upgrading from v1.1.0

While there are a lot of changes between v1.1.0 and v1.1.2, the layout of the configuration has not changed.

Upgrading is a simple as using the install/ update feature from the graphical volume manager or running airboot2.exe.

Upgrading from v1.1.2

The v1.1.4 release was mainly done to implement initial GPT and LUKS awareness and the layout of the configuration has not changed.

Upgrading is a simple as using the install/ update feature from the graphical volume manager or running airboot2.exe.

Downgrading AiR-BOOT

If, for whatever reason, you want to downgrade AiR-BOOT, you'll have to do a remove and reinstall.

First remove AiR-BOOT and then install the version you want to use.

Important: use the installer from the version you are installing !

Removing AiR-BOOT

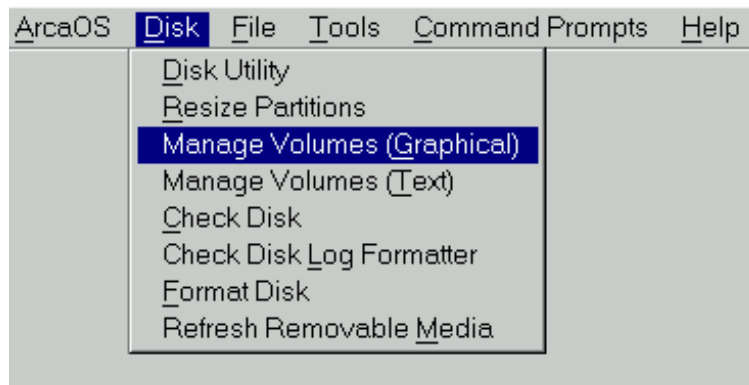
When AiR-BOOT gets installed it replaces the Master Boot Record at the first sector of the disk and puts it's code and configuration at the sectors following the MBR. So, to remove AiR-BOOT, all that is needed is to replace the MBR with the standard OS/2 version. That's what MinilVM does when removing AiR-BOOT.

Removing using the ArcaOS Installation Medium

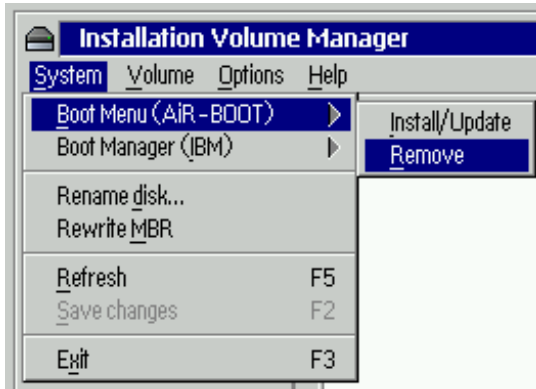
Boot from the ArcaOS Installation Medium and choose to boot from CD.

At the welcome screen, choose System Management.

Once in System Management, start the Graphical Volume Manager.



From the Graphical Volume Manager, choose the following from the menu:



That's all there is to it !

Removing AiR-BOOT using the command-line installer

This is currently not supported.

The reason is that it would involve rewriting the MBR with a "standard" version. However, such a standard version does not exist. Operating Systems like DOS and Windows use their own MBR-code as does OS/2.

Also, such foreign MBR-code cannot be embedded in AiR-BOOT, since it is open-source and this foreign MBR-code is not.

Since booting OS/2 or OS/2 Boot Manager require special Int13X handling when they are located above the 1024-cylinder limit of the BIOS, the safest MBR-code to use is that from the OS/2 v4.5 version. It is also compatible with DOS and Windows. Such an MBR can be generated with the command: `LVM /NEWMBR:1`

Ensure that you use the LVM version from the ArcaOS v5 (or higher) Installation Media for this, otherwise the disk signature in the MBR might get overwritten, most likely causing Windows installations to fail booting.

Removing AiR-BOOT by regenerating the MBR using non-OS/2 systems

Many Operating Systems or disk-maintenance tools incorporate a way to generate a fresh MBR.

If you are using OS/2, using these MBR's might cause some or all of your OS/2 installations to fail to boot. The reason is the absence of an Int13X signature in the MBR, which will cause all installations located above the 1024-cylinder limit to fail booting.

Beware !!

Now that AiR-BOOT is removed, the MBR has been replaced by a standard OS/2 MBR.

The code in this MBR looks for an active primary partition, so don't forget to mark a partition as "startable".

You can use the Installation Volume Manager for that and if you have OS/2 Boot Manager installed that would be the partition to mark startable. Without AiR-BOOT or OS/2-BM GR installed, you cannot boot any logical partitions.

Additional Information

Why OS/2-BMGR and Windows Vista/7 cannot live together

Because the DOS and WINDOWS partitions are primary partitions, OS/2-BMGR hides one when the other is booted from. This is one of the issues with OEM installations of Windows Vista and Windows 7, that use a separate partition for the Windows Boot Manager. When the Windows Boot Manager partition is started from OS/2-BMGR, OS/2-BMGR hides the partition containing Windows Vista/7 causing it to fail booting.

OS/2-BMGR displaying incorrect information

Also note that while the ArcaOS partition is formatted with JFS, OS/2-BMGR shows it as being formatted with HPFS which is incorrect. This is because both HPFS and JFS use the 0x07 partition identifier and OS/2-BMGR has never been updated to recognize JFS.

Installing AiR-BOOT does not remove OS/2 Boot Manager.

Because AiR-BOOT is stored in TRACK0, it does not alter or overwrite OS/2-BMGR, which is stored in it's own partition. AiR-BOOT is able to get control because it modified the MBR that now, instead of booting a partition marked as active, loads the AiR-BOOT program. So, OS/2-BMGR is of no hinder to AiR-BOOT and can easily be re-enabled should you not like AiR-BOOT. In fact, AiR-BOOT is able to boot OS/2-BMGR by chain-loading it, but this is not recommended and the reasons are explained further along in this manual.

AiR-BOOT does not take over any OS/2-BMGR configuration settings

While OS/2-BMGR has some configurable options, like a default system, timeout, etc., AiR-BOOT does not takes these over. AiR-BOOT has it's own configuration, including the above options, and you'll have to set these to your likings.

Advanced Usage

While AiR-BOOT will most likely mainly be used for selecting the operating system to boot, it has some advanced features which users, especially OS/2 developers, might want to put to use.

Some of these advanced features / usages are discussed in the next sections.

Cloning OS/2 Systems

If you are an OS/2 developer, you might want to test some new stuff without changing your current system configuration. Doing a fresh install will take some time and configuring it alike will take some more. With the Drive Letter Feature of AiR-BOOT it becomes more easy to clone a system and even boot it using the same drive-letter.

Preparing the Clone Partition

Create a new partition with MiniLVM and format it with the same file-system as the original.

Assign it a free drive-letter, it does not matter which one.

This is the partition where the clone system will be copied to.

It does not matter if the original is PRI and the clone is LOG, or vice-verse, just as long as the file-system is the same.

If the file-system also differs then you need to copy the corresponding OS2BOOT to the clone.

Also use a partition / volume name to distinguish it from the original.

Creating the Clone

- ◆ Boot from the ArcaOS BOOT-CD to create the clone.
- ◆ Go into Maintenance Mode and verify that both partitions are accessible.
- ◆ Assuming X: is the original and Y: is the clone, give the following command:

```
XCOPY X:\ Y:\ /H /O /T /S /E /R /V
```

This will copy the complete content of the original to the clone, and since you booted from the ArcaOS-CD no files are open.

Reboot to setup AiR-BOOT

Now reboot and enter the AiR-BOOT Setup.

Go to PARTITION SETUP and select the partition of the original.

Now press the L-key and press the letter it boots from, this will force this letter on the original.

Now select the partition of the clone.

Again press the L-key and assign it the *same* drive-letter as the original.

Save the configuration.

Booting the clone

When back in the Main Menu, the clone entry will still show the drive-letter you assigned when you created the partition.

However, when booting it, the drive-letter you forced will be used and the drive-letter on the original will be hidden.

On the next reboot, you will see AiR-BOOT displaying the original with the hidden indicator and both the original and the clone show the same drive-letters. This hiding is not partition-ID hiding but LVM drive-letter hiding !

Booting the original will hide the clone, so on next reboot the clone will be shown as hidden.

Chain-loading OS/2-BMGR

While AiR-BOOT can chain-load OS/2-BMGR, you need to be aware of some things when doing so.

OS/2-BMGR has been around for a long time and in fact was one of the first truly usable boot-managers of it's era.

When OS/2 changed from legacy partitioning to LVM partitioning, OS/2-BMGR also changed in functionality.

On today's systems there is no real need to use OS/2-BMGR and AiR-BOOT is the preferred boot-manager.

However, if you use old OS/2 systems, you might need the OS/2-BMGR that accompanied them since AiR-BOOT might not be compatible with such old systems yet. Booting OS/2-BMGR from AiR-BOOT involves some things you need to be ware of, whether it's a pre-LVM OS/2-BMGR or a recent one.

Set(A)Boot and OS/2-BMGR

OS/2-BMGR is always installed in a primary partition and it gets booted just like any other 'active' partition.

However, OS/2-BMGR has a supporting program called SETBOOT, which is able to manipulate the OS/2-BMGR configuration.

Also, there are various third-party programs that assume that when an OS/2-BMGR partition (0x0A) is present, OS/2-BMGR is in control. This is *not* the case when AiR-BOOT is installed, so manipulating OS/2-BMGR specific stuff by methods other than SETBOOT (AiR-BOOT version) will have no effect.

Relocating GRUB

Relocating GRUB is dependent on the Linux distribution used and the version of GRUB used.

The procedure below is for Ubuntu and it's derivatives like Linux Mint, for other distros consult the accompanied documentation.

These are the steps:

- ◆ Boot the live-CD from your distribution or use a recent Knoppix CD
- ◆ Open a terminal and give the following commands:

```
sudo su
mkdir /1
fdisk -l
mount /dev/sdxn /1
mount -o bind /dev /1/dev
mount -o bind /proc /1/proc
mount -o bind /sys /1/sys
chroot /1
grub-install --force /dev/sdxn
update-grub
exit
umount /1/dev
umount /1/proc
umount /1/sys
exit
reboot
```

The instructions above make you root in the terminal, create a temporary directory to mount the Linux partition, bind the needed virtual file-systems, enter the mounted system, reinstall and update GRUB, exit the mounted system, unmount the virtual file-systems and reboot the system.

The /dev/sdxn is the partition where Linux is installed and where GRUB will be installed after these commands. It can be something like /dev/sda7 or whatnot and fdisk -l or cfdisk will show a list of your partitions.

When AiR-BOOT is installed your Linux System should be in the AiR-BOOT Main Menu and be bootable. Booting it will chain-load GRUB from where you can start Linux just as before the move of GRUB.

Do **not** attempt the above procedure if you are not familiar with the commands given there !

Trouble Shooting

This section contains some tips on trouble shooting with AiR-BOOT.

Overwritten MBR

Since AiR-BOOT is a TRACK0 boot-loader, it is vulnerable to foreign disk management programs that overwrite the MBR. While this does not destroy the AiR-BOOT configuration or it's code, AiR-BOOT is merely disabled. Therefore it's a good idea to have the AiR-BOOT files at hand when such a thing happens. The files you need are: AIRBOOT.BIN, AIRBOOTD.EXE, AIRBOOTW.EXE and AIRBOOT2.EXE. Having these files you are able to repair AiR-BOOT from DOS/Win3x/9x, Windows 2000/XP/Vista/7/8 and OS/2. Simply have them all in the same directory on some medium and run the appropriate installer for the OS you use to repair AiR-BOOT.

Corrupted Code

The AiR-BOOT loader-code is CRC protected.

Every time AiR-BOOT starts it checks the integrity of the loader-code by calculating it's CRC. If the calculated CRC does not match the stored CRC the code is considered corrupted and you will see a screen like this:

```
AiR-BOOT: !ATTENTION!  
- The code of AiR-BOOT is not intact anymore.  
  Please boot via AiR-BOOT disc to restore AiR-BOOT.  
  System halted. Please press RESET._
```

If something like this happens it means that something has altered one or more sectors between the MBR and the AiR-BOOT configuration which is strange since this space is normally not used.

To repair AiR-BOOT just boot from the ArcaOS Installation Medium, open a command-prompt, got to the \SYS\BIN directory and issue the command: AIRBOOT2.EXE. A screen like the one below will be shown:

```
AiR-BOOT Installer v1.1.4 for OS/2  
- (c) Copyright 1998-2012 by Martin Kiewitz.  
  
-> ...Please wait... <-  
- Loading bootcode from file...ok  
- Loading MBR from harddisc...ok  
  
-> ...Current Status... <-  
- AiR-BOOT is not intact  
- Configuration is intact (v1.1.0)  
  
-> ...Please press... <-  
<R> - Repair AiR-BOOT 'v1.1.4/english' on current system  
<D> - Delete AiR-BOOT from current system  
<Q> - Quit without any change
```

As can be seen in ->...Current Status...<-, the installer has determined that the AiR-BOOT code is not intact anymore. It offers a repair option which can be activated by pressing the R-key.

AiR-BOOT User Manual

Version 1.1.4

This will rewrite the loader-code and the following additional lines will be shown:

```
Add/Repair/Update AiR-BOOT in progress...
CB74E5BC
  - Writing AiR-BOOT code...ok

Your copy of AiR-BOOT is now fully functional.
Please hit ESC to exit AiR-BOOT installer or ENTER to reboot your system...
```

Corrupted Configuration

The AiR-BOOT configuration-section is also CRC protected.

So poking values using some sector editor will cause the configuration to become invalid, showing the following screen:

```
AiR-BOOT v1.1.4 - (c) 1998-2017 Martin Kiewitz, Dedicated to Gerd Kiewitz
Build Date: 2017/11/09 01:01:04                                     ArcaOS Edition

Scanning...
- The configuration of AiR-BOOT is not intact anymore.
  Please boot via AiR-BOOT disc to restore AiR-BOOT.
  System halted. Please press RESET._
```

Like with corrupted code, if something like this happens it means that one or more sectors in the AiR-BOOT configuration in TRACK0 have been altered, which should require further investigation.

To repair AiR-BOOT just boot from the ArcaOS Installation Medium, open a command-prompt, got to the \SYS\BIN directory and issue the command: AIRBOOT2.EXE. A screen like the one below will be shown:

```
AiR-BOOT Installer v1.1.4 for OS/2
- (c) Copyright 1998-2012 by Martin Kiewitz.

-> ...Please wait... <-
- Loading bootcode from file...ok
- Loading MBR from harddisc...ok

-> ...Current Status... <-
- AiR-BOOT is intact (v1.1.4)
- Configuration is not intact
- Language is english

-> ...Please press... <-
<R> - Repair AiR-BOOT 'v1.1.4/english' on current system
<D> - Delete AiR-BOOT from current system
<Q> - Quit without any change
```

As can be seen in ->...Current Status...<-, the installer has determined that the AiR-BOOT configuration is not intact anymore. It offers a repair option which can be activated by pressing the R-key.

This will rewrite the default configuration and the following additional lines will be shown:

```
Add/Repair/Update AiR-BOOT in progress...
CB74E5BC
  ▪ Writing AiR-BOOT configuration...ok

Your copy of AiR-BOOT is now fully functional.
Please hit ESC to exit AiR-BOOT installer or ENTER to reboot your system...
```

Because the configuration is rewritten, you lost your settings and have to configure them again using the AiR-BOOT Setup.

OS/2 does not boot

The following sections cover situations where OS/2 fails to boot.

Missing COUNTRY.SYS

If you get the following screen:

```
The system cannot find the file O:\OS2\SYSTEM\COUNTRY.SYS specified
in the COUNTRY statement on line 39 of the CONFIG.SYS file.
Line 39 is ignored.
The system is stopped. Correct the preceding error and
restart the system.
```

Then this means OS/2 was installed on a different drive than the one that is currently set in the LVM information.

This can happen if you are playing with the drive-letter feature or for some other reason.

As you can see the drive-letter where the system expects COUNTRY.SYS is displayed, O: in the case above.

To solve this problem, force this drive-letter on this boot-entry using AiR-BOOT Setup and try to boot the partition again.

This should now succeed. After a successful boot you can remove the forced drive-letter, unless you are using multiple OS/2 installations using this letter.

Unable to operate hard disk

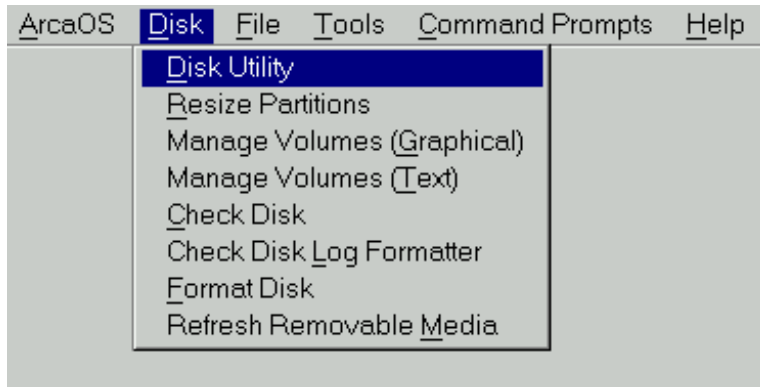
If you get the following screen:

```
OS/2 is unable to operate your hard disk or diskette drive.
The system is stopped. Correct the preceding error and
restart the system.
```

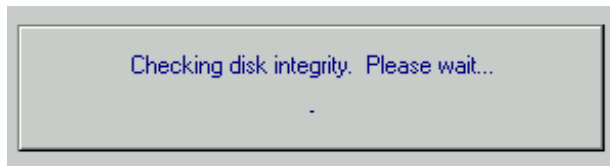
One reason could be the LVM information is missing or corrupted.

This is not something you can fix from within AiR-BOOT and you have to use the ArcaOS Disk Utility.

- ◆ Boot from the ArcaOS CD-ROM
- ◆ Go into Maintenance Mode
- ◆ Choose Disk-Disk Utility from the Menu

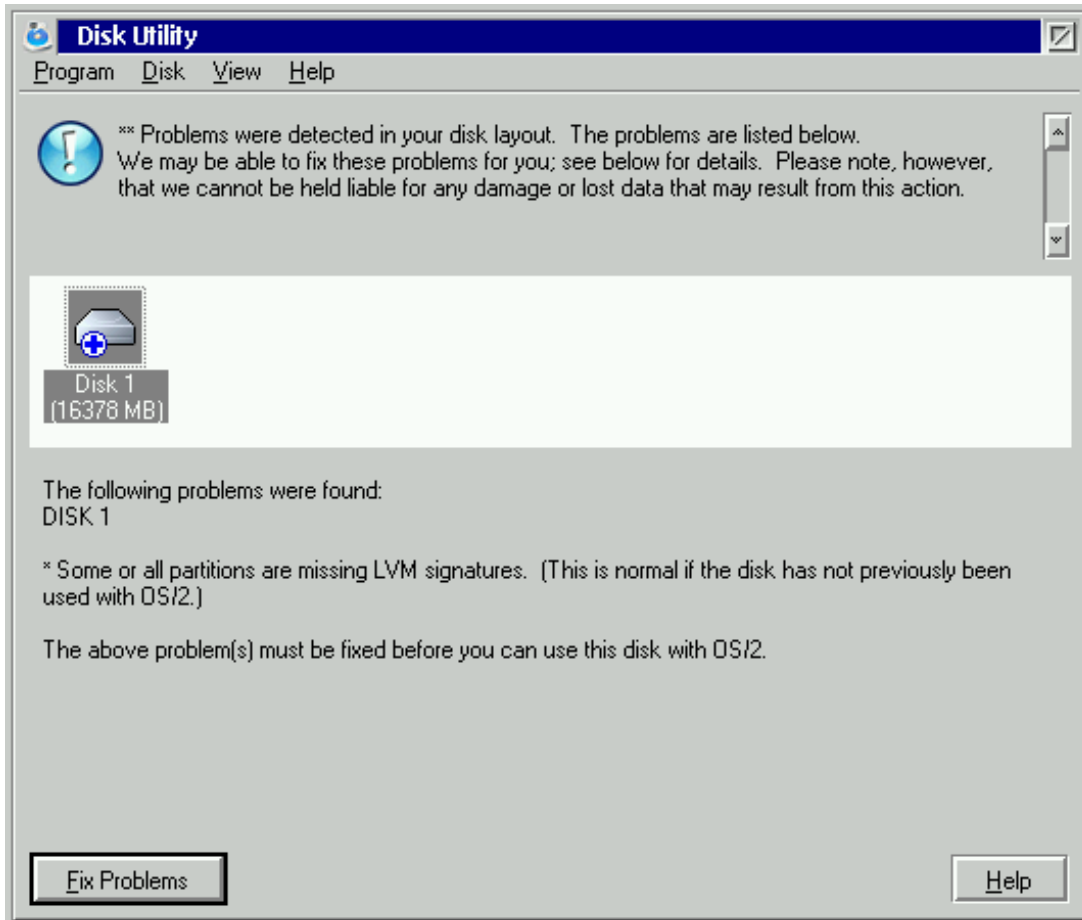


Wait for the integrity check to complete.

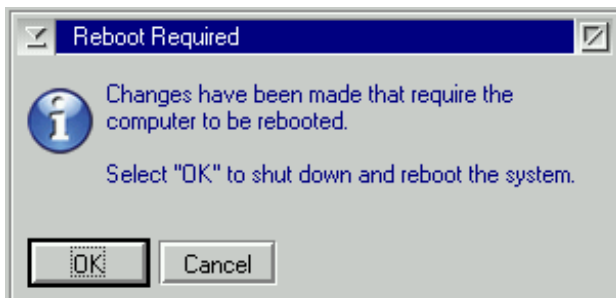


- ◆ The Disk Utility window will appear
- ◆ Now click on the disk in question and click on the *right* mouse-button

You will see a screen like below.



Now click on the Fix Problems button and wait for the dialog below to appear:



Do not reboot yet!

Because the LVM-information was absent and has been regenerated, there is no drive-letter associated with the partition yet. This can be rectified by now starting the Graphical Volume Manager, selecting the partition and assigning it the proper drive-letter and label. The drive-letter to assign must correspond to the drive-letter used in the CONFIG.SYS of the partition.

After setting the correct drive-letter, rebooting from the hard-disk should now succeed.

Virus warning

When you have setup AiR-BOOT to detect viruses, it checks the CRC of the Partition Boot Record against the stored CRC in the AiR-BOOT configuration. If there is a mismatch a screen resembling this will appear:

```
AiR-BOOT v1.1.4
(c) 1998-2017 Martin Kiewitz, Dedicated to Gerd Kiewitz
This is GPLv3+ software, please visit: http://www.gnu.org/licenses/gpl.txt
To obtain the sources, please visit: http://svn.netlabs.org/air-boot

Booting the system using ArcaOS partition; please wait...

- !ATTENTION! -> A VIRUS WAS FOUND <- !ATTENTION!
  It is located in the boot-record of the partition, you wanted to boot.
  Use a virus-checking program. It could be false alarm either.
  After removal, you have to reinit the detection variables, go into
  'PARTITION SETUP' and switch VIBR-detection two times (off/on).
  If this was just a false alarm, leave it in off-state.
  System halted. Please press RESET.
```

Of course this could be a valid virus warning.

However, PBR viruses are not so common anymore and a more likely cause for this warning is that some program altered the PBR.

You cannot boot this partition until you clear its 'infected' state:

- ◆ Enter the AiR-BOOT Setup
- ◆ Enter PARTITION SETUP
- ◆ Place the selection bar on the partition in question
- ◆ Press the V-key *two* times

This clears the infected state of the partition.

Third Party Tools

There may exist a situation where you need to modify the layout of your disks, or fix problems, which are beyond the possibilities offered by the OS/2 Logical Volume Management tools. This is especially the case when reorganizing a huge hard-disk (>512GB) to enable installation of OS/2 in the lower 512GB region.

Note that using third party tools, especially those that are *not* OS/2 aware, is not recommended. For instance, an alternative to moving partitions is to backup a partition on the file level using:

```
XCOPY X:\ Y:\ /H /O /T /S /E /R /V
```

then do the repartitioning and use the same command to restore the data.

DFSee from FSys Software

The DFSee program has been a long time companion of advanced OS/2 users.

It is your Swiss Army Knife for managing OS/2 partitioned disks and it offers a multitude of powerful features you can use to protect or rescue data. More information can be found on its web-site:

<https://www.dfsee.com>

It is really worth considering to obtain a license for the DFSee program as it can be a valuable tool to protect or rescue your data.

The SystemRescueCD

The SystemRescueCD boots a Gentoo based Linux system and offers the GParted tool.

While its main focus is on Linux disk management, it can be used to do complex operations like moving entire partitions. However, the tools provided are not OS/2-LVM aware and after using them the regular OS/2 LVM tools should be used to regenerate any LVM information that got lost in the process. More information can be found on its web-site:

<http://www.system-rescue-cd.org>

When using GParted, do *not* forget to enable Cylinder Alignment *before* doing any operation !

Caution !!

This section is about some things you should keep in mind. Some are related to using AiR-BOOT, others may not.

Installation of Windows Vista / 7 and 8(?) overwrites primary partitions with type 0x07

The installation of Windows Vista and Windows 7 has been greatly simplified with regard to Windows XP. Not only with regard to the user but apparently also with the regard to check the file-system used by 0x07 partitions.

Since type 0x07 is an installable file-system indicator, and has been introduced in OS/2 v1.1 which supported HPFS, it does not specify a specific file-system or operating-system at all. The OS/2 HPFS, OS/2 JFS, Windows NTFS and Windows exFAT all use this 0x07 identifier.

If you have an OS/2 installation in a primary partition using HPFS or JFS, and thus using the type 0x07 identifier, installation of Windows Vista or Windows 7 can potentially *overwrite* the partition without any warning !

Depending on the layout of your hard disk, the Vista / 7 / 8(?) installation program presents these partitions as unused or usable for the Windows installation you are doing without correctly identifying on which partition it will be installed.

The installable file-system is an invention of Microsoft itself and even a humble program like AiR-BOOT can detect which file-system the 0x07 partition uses.

Therefore this behavior is considered a *deliberate* act to destroy non-Windows systems on the users hard disk.

I wonder if they also overwrite their own exFAT data which also uses type 0x07.

Using AiR-BOOT on USB sticks

Whether AiR-BOOT can be used on USB sticks is highly dependent on how the BIOS handles USB Mass Storage devices and physical disk numbering. Because USB sticks can be put into any system able to boot from them, the functionality of AiR-BOOT is dependent on the system that boots the stick. If the BIOS does not handle disk-numbering the way AiR-BOOT expects, it may overwrite the MBR and possibly TRACK0 of the real hard disk instead of the booted USB stick !

It is not advised to use AiR-BOOT on USB unless you know for sure it works on the systems you boot from it !

Using multiple DOS installations can boot wrong partition

DOS was never designed to boot with other primary partitions visible and formatted with FAT12 or FAT16.

More specifically, the DOS PBR boot-code just boots the first primary FAT-partition it sees (and is bootable).

When you have two DOS systems installed, on the first and second primary partitions, and you boot the second one, changes are that the PBR code actually boots the first partition.

This goes even more for older DOS versions, like the v3.x series.

You can work around this by using the Partition Hiding feature of AiR-BOOT to hide all other partitions for a specific DOS install.

FreeDOS is more flexible and can even be installed in a logical partition, but partition hiding is still needed because also the PBR code of FreeDOS plays some tricks.

AiR-BOOT User Manual

Version 1.1.4

The AiR-BOOT 45 Partitions Limitation

AiR-BOOT currently supports up to 45 partitions.

This means *all* the partitions found (including non-bootable ones) on *all* disks !

Future versions might lift this limitation.

When more than 45 partitions are found AiR-BOOT starts with a warning notice and halts for you to read it.

Pressing ENTER makes AiR-BOOT continue, but the last partitions will not be accessible !

To indicate this exceptional situation during AiR-BOOT usage, the selection-bar is colored red.

You should correct this situation as soon as possible !

The Warning Pop-up when using >45 Partitions



Too many partitions found. AiR-BOOT is supporting up to 45.

The Main Menu with the red selection-bar when using >45 Partitions

No	Hd	Label :	Type :
01	01	DOS	FAT16Big
02	01	WINDOWS	NTFS
03	01	ArcaOS	JFS on drive 0:
04	02	toomany-00	FAT16Big
05	02	toomany-01	FAT16Big
06	02	toomany-02	FAT16Big
07	02	toomany-03	FAT16Big
08	02	toomany-04	FAT16Big
09	02	toomany-05	FAT16Big
10	02	toomany-06	FAT16Big
11	02	toomany-07	FAT16Big
12	02	toomany-08	FAT16Big
13	02	toomany-09	FAT16Big
14	02	toomany-10	FAT16Big

Timed boot disabled; no timeout will occur.

Press [Esc] to toggle timed boot, [Enter] to accept current selection.
Select another with the arrow keys, or press [TAB] to see BIOS POST message.
[DEL] to Power Off [F10] to enter Setup

Quick FAQ

This section attempts to answer some frequently asked questions.

Where is AiR-BOOT stored ?

In TRACK0 of the boot-disk.

AiR-BOOT is a so called TRACK0 boot-loader and does not require a dedicated partition like OS/2-BMGR.

However, the space in TRACK0 is a "nowheres land" and besides the MBR the rest of the space has no official designation.

Therefore, the space may be claimed by other systems like the Ontrack Disk Manager.

The last sector of TRACK0 is used by the OS/2 LVM system to store information about the disk geometry and the four partition entries that the MBR can contain. AiR-BOOT of course recognizes this usage.

The layout of TRACK0 when AiR-BOOT is installed is roughly:

- ◆ AiR-BOOT MBR
- ◆ AiR-BOOT Loader Code
- ◆ AiR-BOOT Main Configuration
- ◆ AiR-BOOT Internal Tables
- ◆ Backup of the MBR last booted
- ◆ LVM sector (not part of AiR-BOOT)

Can AiR-BOOT boot Linux ?

Yes, by chain-loading the boot-loader used by the Linux installation.

For this to work the Linux boot-loader, usually GRUB, must be installed in the partition where Linux is installed and not in the MBR space. If you currently have GRUB installed in the MBR you need to relocate it.

I have more Windows Vista / 7 installations and they are all called Windows... (?)

Windows uses it's own PBR boot-loader which can be installed in a DOS partition, a partition it bluntly overwrote that contained your OS/2 installation or an NTFS partition where also the Windows system itself is installed. With one Windows version installed this loader is invisible because it just boots this only Windows system. With multiple Windows installations however, you are presented with a menu and the fact that they are all called Windows is a feature of the Windows boot-loader. You should know by know that in Microsoft's mind there exist no other operating systems besides Windows. I have found no user friendly way to alter the names presented, but I did not expect that from a brain-dead loader such as this. Call the Microsoft tech support and after that contact their legal department to notify them of your intention to file a class-action law-suit for deliberately destroying OS/2 installations when installing Windows.

Can AiR-BOOT boot ancient OS/2 systems ?

Yes, it most probably can.

You may need to use the Partition Hiding feature to be successful.

Also, you could install the OS/2 Boot Manager of the ancient system in question and chain-load that.

Beware that ancient systems have severe limitations with regard to disk sizes and are often bound to the famous BIOS 1023 cylinder limit.

Best is to keep such old systems below the 8 GiB boundary.

Does installing AiR-BOOT destroy OS/2 Boot Manager ?

No.

It just takes over the MBR that now loads AiR-BOOT instead of loading the PBR of the active partition (OS/2-BMGR).

You can even chain-load your OS/2-BMGR from within AiR-BOOT but use that with caution.

Simply giving the command LVM /NEWMBR:1 in an OS/2 prompt and then setting Boot Manager active will restore OS/2-BMGR as your boot-manager.

AiR-BOOT versions

The following sections cover the versions released since v1.06.

The AiR-BOOT version that comes with ArcaOS is specifically modified to replace the functionality of OS/2 Boot Manager and to cope with other issues like huge (>502GiB) disks. Below is a more thorough explanation.

AiR-BOOT classic

All AiR-BOOT versions in the v1.x.y series are based upon the v1.06 code-base.

This implies that they are TRACK0 bound, written in Assembly and will never exceed 31KiB in size.

Version 1.1.4

This version is included with the release of ArcaOS v5.0.2 in November 2017.

It contains some enhancements with regard to GPT partitioned and LUKS formatted disks.

Below is a short list of changes, some user some developer related:

- ◆ Updated the manual to reflect v1.1.4
- ◆ Skip GPT partitioned and LUKS formatted disks
- ◆ Removed Cooper Bars to create space for future enhancements

Version 1.1.2

This version included with the release of ArcaOS v5.0 in May 2017.

It is more aware of removable media like USB sticks and the LVM logic has received a major overhaul.

Below is a short list of changes, some user some developer related:

- ◆ Updated the manual to reflect v1.1.2
- ◆ Enhanced the transformation from odt to inf
- ◆ Added building the manual to the build-system
- ◆ Enhanced the overall build-system
- ◆ All sources lowercased
- ◆ Fixed building on linux
- ◆ Introduction of file-commander user-menus for building
- ◆ Updated contact information
- ◆ Replaced references to ecs with more generic os/2
- ◆ Enabled soft reset when airboot halts due to an error
- ◆ Removed many hard-coded 80h bios-disk numbers, enabling >80h boot-disk
- ◆ Improved int13x functionality
- ◆ Completely revamped the lvm code
- ◆ Improved the handling of removable media
- ◆ Initial awareness of large-floppy formatted removable media
- ◆ New pre-boot summary screen

Version 1.1.0

This is the version as released with eComStation v2.2-beta in may 2013.

Below is a short list of changes, some user some developer related:

- ◆ Fixed 45 partitions bug
- ◆ Fixed booting win7 from fat32
- ◆ Fixed hpfs zero-drive issue
- ◆ Converted sources for use with jwasm
- ◆ Fixed bogus virus warning

Version 1.0.8

This version was never released.

It has been a development version all the way between the release of v1.07 and v1.1.0.
Some interim release candidates have been made available for testing purposes.

All v1.0.8 test-versions built before 13 May 2012 contain a severe bug that can wipe the MBR of *all* hard-disks in the system !!
Any version of AiR-BOOT v1.0.8 should not be used at all and deleted from the face of the Earth !

Version 1.07

Version v1.07 was the first OS/2 enhanced version and was delivered with the release of eComStation v2.1 in may 2011.
It contained the wrong contact information and many people used the source-forge site to post issues.

Below is a short list of changes, some user some developer related:

- ◆ Added support for hard-disks larger than 512GB
- ◆ Added support for suppressing main menu when doing a fresh OS/2 installation
- ◆ Increased number of supported partitions from 30 to 45 (buggy in this version)

This version also contains a number of known bugs, which are:

Force BIOS-LBA Usage must be *enabled*.

Do *not* disable this feature when using v1.07 !

When disabled, AiR-BOOT uses CHS-addressing and with todays large hard-disks this can lead to troubles.

Drive Letter Feature is broken

The drive-letter feature is broken in that it does not remember the assigned letters on the next boot.

It is even possible that a non-printable character now represents the drive-letter.

This can cause OS/2 to fail to boot.

Do not use this feature with v1.07 !

Partition Hiding could fail

This bug and the Drive Letter Feature bug are caused by the extension made to AiR-BOOT v1.07 to support 45 partitions instead of 30.

However, this was not done correctly and using a number of partitions that exceeds 30 and / or hiding more than 29 partitions can cause a table overflow and AiR-BOOT to malfunction.

Booting Windows sometimes failed

Due to a bug in the preparation of the in-memory passed PBR, some Windows installations failed to boot when the Windows boot-loader was on a FAT32 partition. This has been fixed.

Version 1.06

This is the last version developed by Martin Kiewitz.

The OS/2 enhanced versions, starting with v1.07, are based upon these sources but are not developed / maintained by Martin.

AiR-BOOT reloaded

There is another AiR-BOOT project in the works that will address things like EFI and GPT and other stuff needed to keep OS/2 installable and bootable on modern hardware. The reloaded version is a rewrite from scratch with the classic version being its inspiration. It will probably have a 2.x.y versioning scheme. Don't expect it to be available tomorrow since modern developments like EFI/GPT are not trivial things to implement with regard to OS/2.

Some preliminary information

The following sections are subject to change or removal and should not be interpreted as specifications or definitely available features of the reloaded version.

Can be installed in a dedicated partition

With the GPT partitioning scheme there is no concept of Cylinder, Head and Sector anymore, so a concept of "TRACK0" does not exist. Besides that, the space that AiR-BOOT currently uses is being used by the GPT partition-table and is thus unavailable anyway. It is intended that such a dedicated partition can be MBR or GPT based.

Modular Design

The classic version has a monolithic design and can not easily scale, either due to its monolithic nature or to the lack of code-space. Using components with a well defined interface makes it possible to more easily scale or morph to adapt to certain environments. An example of such a component would be the user interface, where a menu driven one can be used when space allows or a simple 1,2,3 interface when space is severely limited. Both menu components would export the same interface and just differ in the implemented functionality.

Main logic flow in C++

To accomplish a modular and flexible design, object orientation together with abstraction is needed, so the main flow of things will be in C++ to easily glue stuff together. Methods can be implemented in tightly coupled Assembler to retain code compactness and low level control.

Be configurable by external tools by exporting a configuration interface

Such an interface will export an API that can be used by external tools, like DFSee and the WPS, without having to know the internal details of how and where the configuration is stored.

Handle the new EFI and GPT technologies

The BIOS is on the verge of being replaced with EFI and the MBR partitioning scheme with GPT. These developments pose a number of problems with regard to booting OS/2 and it could be helper-stuff will be implemented in AiR-BOOT to overcome these issues. At the time of writing this is still much a gray area.

Don't hold your breath

Support for EFI and GPT are not trivial things and it may as well be that modifying something like GRUB proves to be an easier way to support OS/2 in the future, rendering the whole AiR-BOOT Reloaded idea redundant. Time will tell...

Feedback

Today's systems are much more complex than the simple IBM XT/AT architecture with which the PC era began. Besides numerous enhancements in disk-interfaces, local buses and whatnot, modern systems are capable of advanced things like swapping disks, booting from USB and other storage devices. Even the BIOS is gradually being replaced by UEFI and the familiar MBR partitioning scheme is being phased out in favor of the GPT partitioning scheme. In light of these things you are very much encouraged to provide feedback to help further development of AiR-BOOT or any other component of OS/2 for that matter.

Using the AiR-BOOT Trac Project Page

AiR-BOOT is released under the GPL v3+ so it's sources are freely available.

Please note that the version of AiR-BOOT that is distributed with ArcaOS is a specially adapted version and it can be considered a fork. While it is based on the v1.06 sources that reside at sourceforge, the sources for this OS/2 adapted version reside at netlabs. Therefore, please do not report issues with v1.07 or higher at the sourceforge site, but use the links below.

The AiR-BOOT Trac Project Page is at: <http://svn.netlabs.org/air-boot>

Browsing the sources

If you want to browse the sources on-line, then use this link: <http://svn.netlabs.org/air-boot/browser/trunk>

Getting the sources

For this you will need an SVN-client v1.5.x or higher.

Create a directory air-boot, change to it and then give the following command:

```
svn checkout http://svn.netlabs.org/repos/air-boot/trunk trunk
```

This will create a directory trunk that contains the master Makefile and build-scripts.

Reporting issues

As stated your feedback is very important, so if you encounter bugs or other issues it is very much appreciated if you report them by creating and submitting a trac-ticket. This is the link to do so: <http://svn.netlabs.org/air-boot/newticket>

Contributing

Besides providing feedback, there are other ways to contribute to AiR-BOOT.

If you have skills with regard to artwork, a nice theme of icons and other bitmaps would be nice.

Reviewing documentation or contributing paragraphs or chapters you think may be of value are also highly appreciated.

And of course any code fixes and / or enhancements are also welcome.

Using the Arca Noae BugTracker

This is another way of reporting bugs with regard to ArcaOS components.

They are all registered in a database and then assigned to a developer by a manager.

To report a problem or issue using the Arca Noae bug-tracker use this link: <https://mantis.arcanoae.com>

Note however that reporting using Trac is preferred because it is project-bound.

Contact using E-Mail

Besides problems or other issues, you might have ideas for enhancements or whatever.

Then you can send an e-mail to: rousseau.os2dev@gmx.com

Special Thanks

The following people deserve special thanks.

Martin Kiewitz

For developing AiR-BOOT in the first place.

For making it open-source.

For all his other contributions to the OS/2 realm, and

for even responding to problems with the v1.07+ versions which are not by his hand.

AiR-BOOT is a "Piece of Art" and is just what OS/2 needed to be able to cope with new systems, huge hard-disks and other recent developments.

It's a "source of inspiration" and future versions will be built upon it's initial concept.

Steven Levine

For his help with tracking down some 'unwilling' bits, the vast knowledge he so kindly shares,

his work on USB and many other projects, his unmissable problem tracking skills

and his valuable "I would do it like this" input.

Alex Taylor

For implementing AiR-BOOT support in MiniLVM.

His awesome work on taming LVM and all his other contributions to the OS/2 realm.

David Azewerick

For taking the time to tackle a nasty problem with AiR-BOOT and HPFS.

For his much appreciated feedback on technical issues,

and the wonderful work he is doing on ACPI and the other projects he works on.

Barry Landy

For fearlessly testing several beta-versions,

discovering some severe bugs and providing useful suggestions.

The unmentioned

Everyone else who provided valuable feedback to make AiR-BOOT a better product.

Disclaimer

The current maintainer is in no way responsible for any damage that AiR-BOOT may inflict on your system.

If it fries your cpu, blows up your monitor or changes the layout of your keyboard to Dvorak when you're not looking, it's all your fault.

By installing and using AiR-BOOT you acknowledge this and know that you and only you are responsible.

With that out of the way, you are of course not left in the dark where "Nobody can hear your scream".

Contact rousseau.os2dev@gmx.com and a sincere effort will be made you get you out of the dark.

Ben Rietbroek,

AiR-BOOT Maintainer.