Computer Hacking Forensic Investigator

Module IX

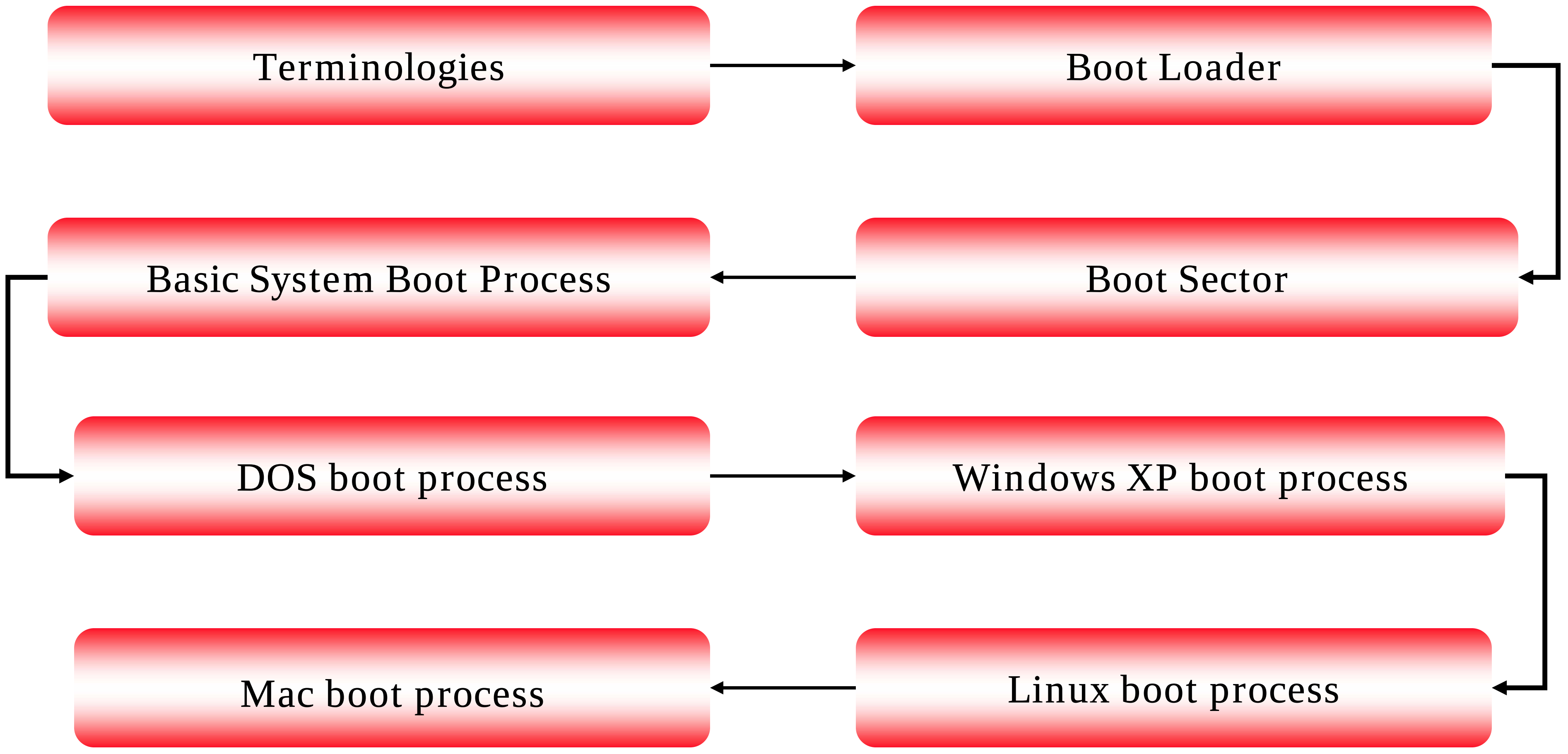
Windows, Linux and Macintosh Boot Process

Module Objective

This module will familiarize you with the following:

- Terminology
- Boot loaders
- Boot sectors
- Basic system boot process
- MSDOS boot process
- Windows XP boot process
- Linux boot process
- Macintosh boot process

Module Flow



EC-Council

Terminologies

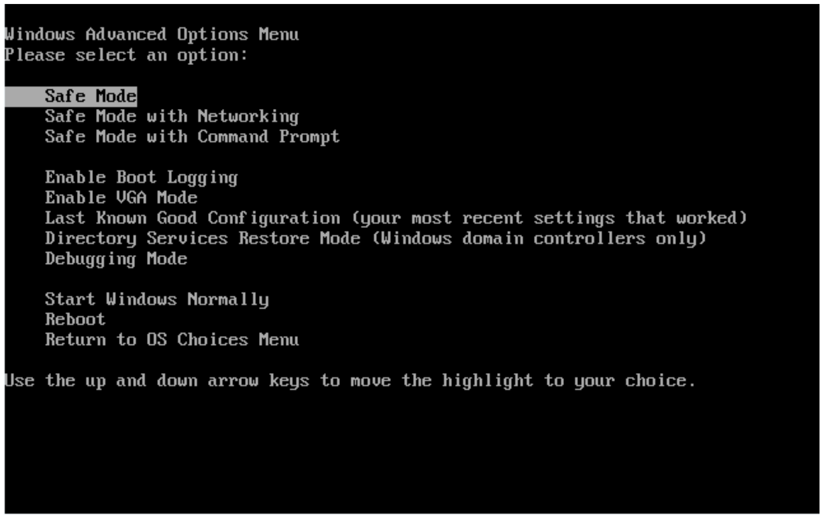
- **Booting**
  - Booting is a process that starts operating systems when the user turns on a computer system.
- **Bootstrap**
  - Bootstrap may be defined as a simple program that actually begins the initialization of the computer's operating system.
- **BIOS**
  - Basic Input/Output System or Basic Integrated Operating System.
  - Performs booting process.
- **CMOS**
  - Complementary metal oxide semiconductor (CMOS) is a widely used type of semiconductor.
  - Computers contain a small amount of battery-powered CMOS memory to hold the date, time, and system setup parameters.
- **Boot Sequence**
  - It is the set of operations the computer performs when it is switched on that load an operating system.
- **Cold boot (Hard boot)**
  - Starting computer from a powered-down, or off, state.
- **Warm boot (Soft boot)**
  - Restarting computer that is already turned on via the operating system.

Boot Loader

- It is a small program that loads the operating system into the computer's memory when the system is booted.

- A basic bootloader has following eight instructions:

```
0: set the P register to 8.
1: check paper tape reader
   ready.
2: if not ready, jump to 1.
3: read a byte from paper tape
   reader to accumulator.
4: if end of tape, jump to 8.
5: store accumulator to address
   in P register.
6: increment the P register.
7: jump to 1.
```



```
Windows Advanced Options Menu
Please select an option:

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable VGA Mode
    Last Known Good Configuration (your most recent settings that worked)
    Directory Services Restore Mode (Windows domain controllers only)
    Debugging Mode

    Start Windows Normally
    Reboot
    Return to OS Choices Menu

Use the up and down arrow keys to move the highlight to your choice.
```

Boot Sector

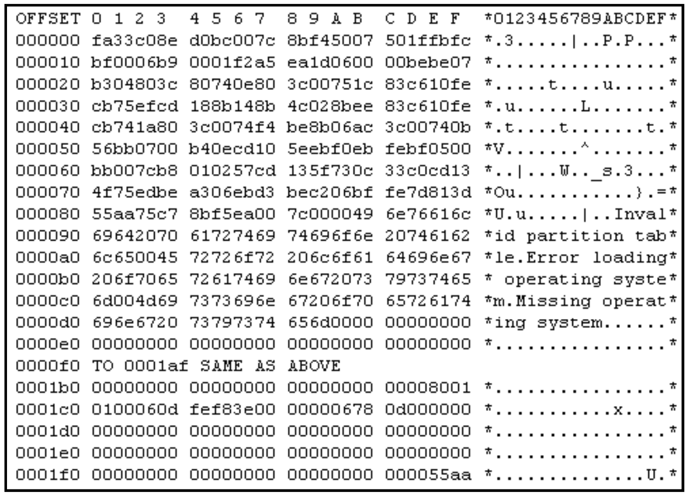
- A boot sector is a sector of a hard disk, floppy disk, or similar data storage device that contains code for bootstrapping programs.
- Each valid boot sector has two bytes (0x55AA), called a boot sector signature, at the end of the sector.
- There are two major kinds of boot sectors:
  - Volume Boot Record
    - It is first sector of a data storage device that has not been partitioned or first sector of an individual partition on a data storage device that has been partitioned.
    - It contains code to load and invoke the operating system or other standalone program installed on that device or within that partition.

Boot Sector (cont'd)

- Master Boot Record
  - It is first sector of a data storage device that has been partitioned.
  - It contains code to locate the active partition and to invoke its Volume Boot Record.
  - Master boot record contains the following structures:
    - Master Partition Table

      It is a small bit of code, referred as a table, that contains a complete description of the partitions that are contained on the hard disk.
    - Master Boot Code

      The master boot record is the small bit of computer code that the BIOS loads and executes to start the boot process.

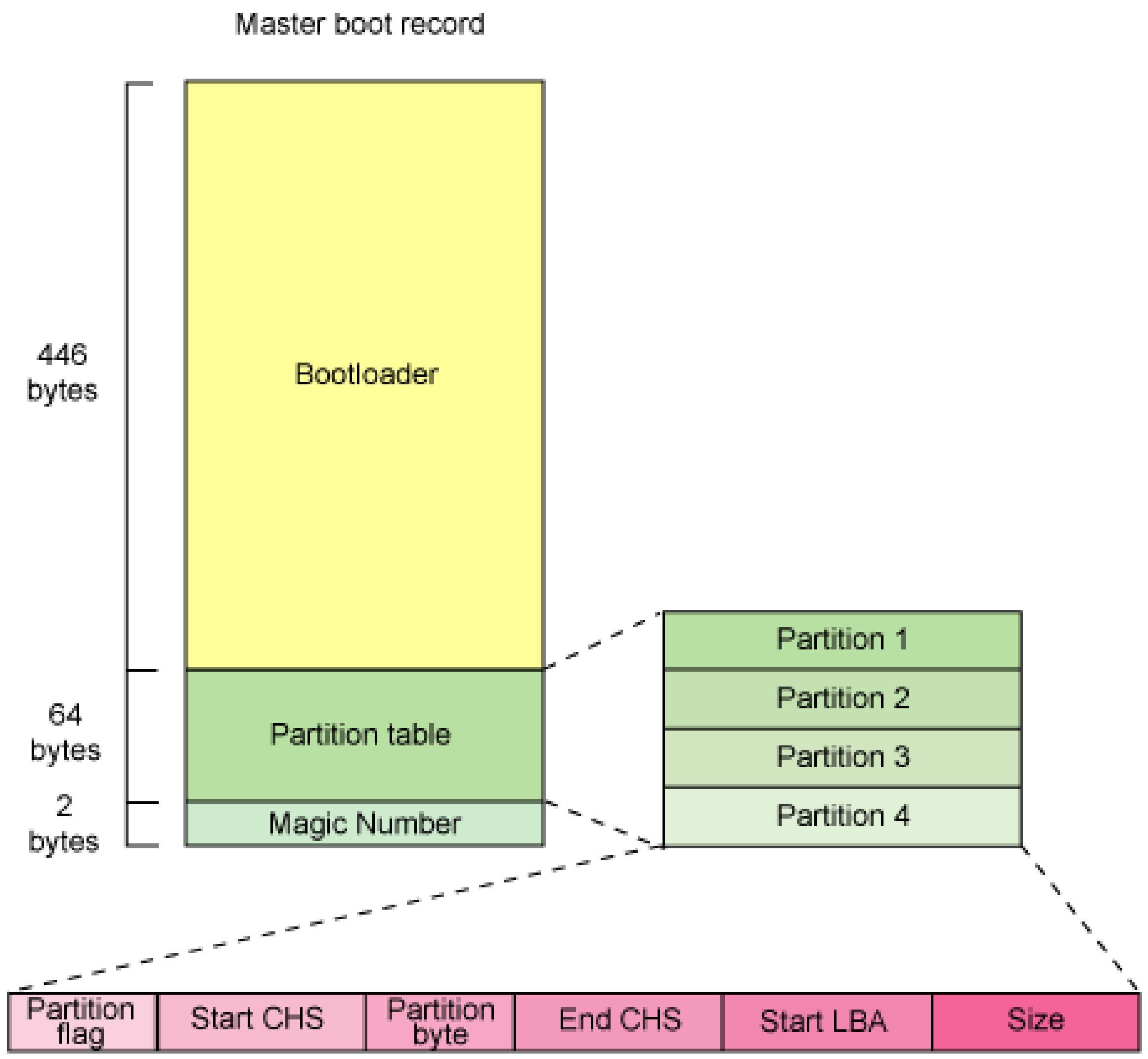


```
OFFSET 0 1 2 3  4 5 6 7  8 9 A B  C D E F  *0123456789ABCDEF*
000000 fa33c08e d0bc007c 8bf45007 501ffbfc *.3.....|..P.P...*
000010 bf0006b9 0001f2a5 ea1d0600 00bebe07 *................*
000020 b304803c 80740e80 3c00751c 83c610fe *.....t....u.....*
000030 cb75efcd 188b148b 4c028bee 83c610fe *.u......L.......*
000040 cb741a80 3c0074f4 be8b06ac 3c00740b *.t....t.......t.*
000050 56bb0700 b40ecd10 5eebf0eb febf0500 *V.......^.......*
000060 bb007cb8 010257cd 135f730c 33c0cd13 *..|...W.._s.3...*
000070 4f75edbe a306ebd3 bec206bf fe7d813d *Ou..........}.=*
000080 55aa75c7 8bf5ea00 7c000049 6e76616c *U.u.....|..Inval*
000090 69642070 61727469 74696f6e 20746162 *id partition tab*
0000a0 6c650045 72726f72 206c6f61 64696e67 *le.Error loading*
0000b0 206f7065 72617469 6e672073 79737465 * operating syste*
0000c0 6d004d69 7373696e 67206f70 65726174 *m.Missing operat*
0000d0 696e6720 73797374 656d0000 00000000 *ing system......*
0000e0 00000000 00000000 00000000 00000000 *................*
0000f0 TO 0001af SAME AS ABOVE
0001b0 00000000 00000000 00000000 00008001 *................*
0001c0 0100060d fef83e00 00000678 0d000000 *..........x....*
0001d0 00000000 00000000 00000000 00000000 *................*
0001e0 00000000 00000000 00000000 00000000 *................*
0001f0 00000000 00000000 00000000 000055aa *..............U.*
```

MBR record in hex and ASCII

EC-Council

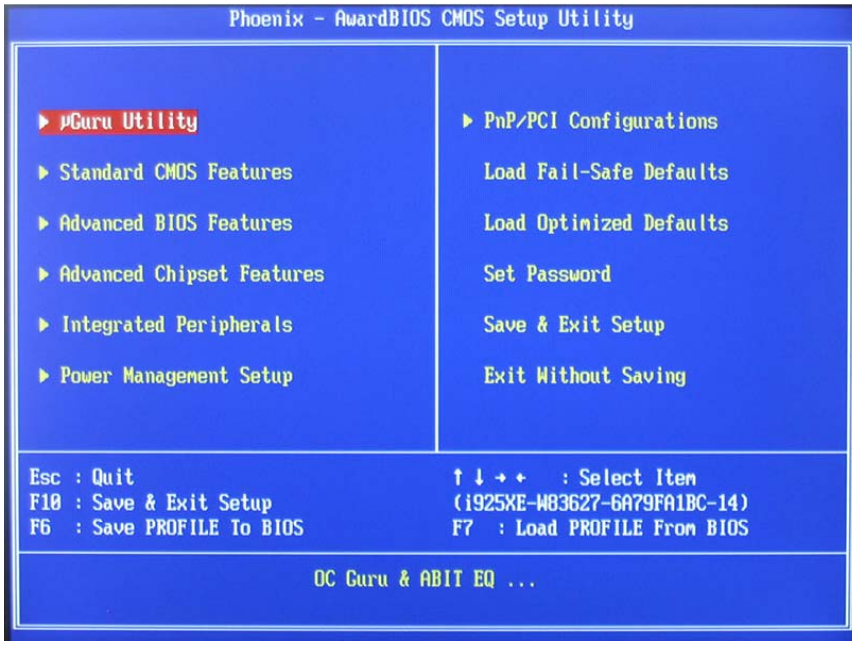
Anatomy of MBR



Basic System Boot Process

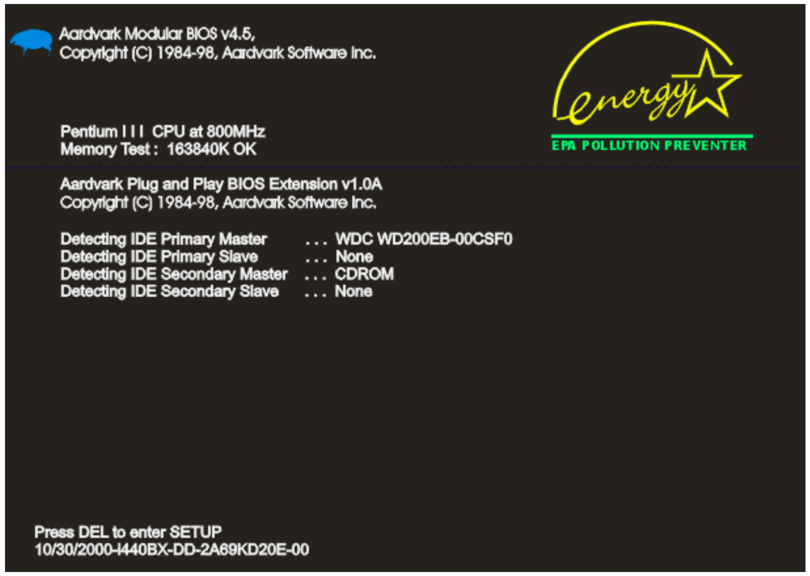
After the computer's power is turned on:

- The system clock generates a series of clock ticks which initializes CPU.
- CPU looks to the system's startup program in the ROM BIOS for its first instruction.
- First instruction is to run the power-on self test (POST), in a predetermined memory address.
- POST checks the BIOS chip and then tests CMOS RAM.



Basic System Boot Process (cont'd)

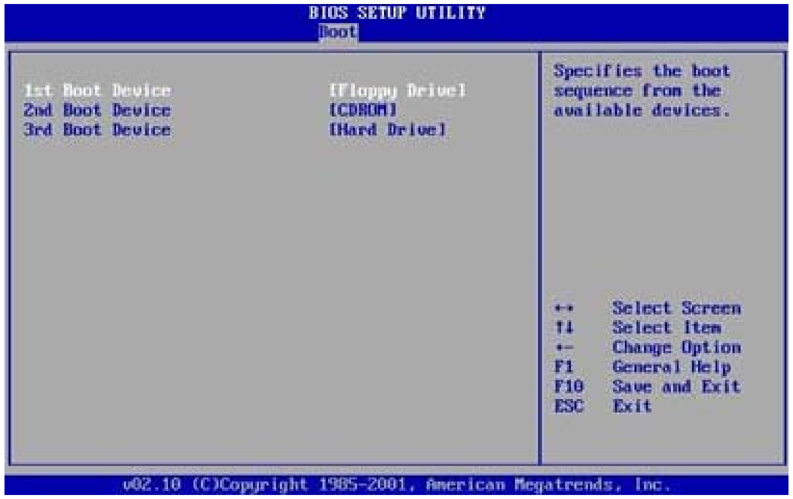
- If there is no battery failure, POST checks the inventoried hardware devices (video cards), secondary storage devices (hard drives and floppy drives), ports and other hardware devices, (keyboard and mouse) to ensure they are functioning properly.
- CPU initialization is completed if everything is fine.
- BIOS looks in to the CMOS chip to find the drive where OS is installed.



```
Aardvark Modular BIOS v4.5,
Copyright (C) 1984-98, Aardvark Software Inc.


Pentium I I I  CPU at 800MHz
Memory Test :  163840K OK

Aardvark Plug and Play BIOS Extension v1.0A
Copyright (C) 1984-98, Aardvark Software Inc.

Detecting IDE Primary Master      . . . WDC WD200EB-00CSF0
Detecting IDE Primary Slave       . . . None
Detecting IDE Secondary Master    . . . CDROM
Detecting IDE Secondary Slave     . . . None


Press DEL to enter SETUP
10/30/2000-I440BX-DD-2A69KD20E-00
```

Basic System Boot Process (cont'd)

- BIOS then checks the boot record of drive to find the beginning of the OS and the subsequent program file that initializes the OS.
- BIOS copies its files into memory after OS initialization.
- OS takes over control of the boot process.
- OS performs another inventory of the system's memory and memory availability and loads the device drivers that it needs to control the peripheral devices, such as a printer, scanner, optical drive, mouse and keyboard.



```
                    BIOS SETUP UTILITY
                         Boot

1st Boot Device          [Floppy Drive]      Specifies the boot
2nd Boot Device          [CDROM]             sequence from the
3rd Boot Device          [Hard Drive]        available devices.

                                             ↔    Select Screen
                                             ↑↓   Select Item
                                             +-   Change Option
                                             F1   General Help
                                             F10  Save and Exit
                                             ESC  Exit

     v02.10 (C)Copyright 1985-2001, American Megatrends, Inc.
```

EC-Council

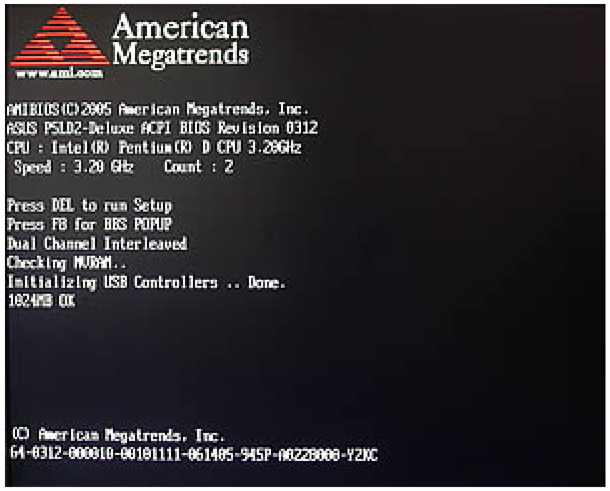
MS-DOS Boot Process

Microsoft
MS-DOS
Operating System

MS-DOS Boot Process

After the computer’s power is turned on:

- BIOS initiate Power On Self Test (POST). It performs following actions:
    - Initializes keyboard and other system hardware, such as the video card.
    - Tests the computer's RAM memory.
    - Checks disk drives.
    - Attempts to find a valid operating system.



```
AMIBIOS(C)2005 American Megatrends, Inc.
ASUS P5LD2-Deluxe ACPI BIOS Revision 0312
CPU : Intel(R) Pentium(R) D CPU 3.20GHz
 Speed : 3.20 GHz    Count : 2

Press DEL to run Setup
Press F8 for BBS POPUP
Dual Channel Interleaved
Checking NVRAM..
Initializing USB Controllers .. Done.
1024MB OK


 (C) American Megatrends, Inc.
64-0312-000010-00101111-061405-945P-A0228000-Y2KC
```

MS-DOS Boot Process (cont'd)

If all goes well in POST test,

- BIOS search for the File IO.SYS (INPUT OUTPUT SYSTEM) in the ROOT Directory and loads it into RAM.
- IO.SYS provides services for peripherals such as printers, modems, disk drives, etc.
- IO.SYS loads file called MSDOS.SYS which is an extension of ROM or BIOS.
- MSDOS.SYS acts as a gateway for the communication between DOS and BIOS.

MS-DOS Boot Process (cont'd)

- MSDOS.SYS searches and loads CONFIG.SYS file.
- CONFIG.SYS stores system configurations.
- CONFIG.SYS is processed and executed.
- MSDOS.SYS loads COMMAND.COM program file into memory.
- COMMAND.COM is a command Interpreter which contains all the internal DOS instructions such as DIR, CD, CLS, etc.
- COMMAND.COM, finally, searches and loads another optional file called AUTOEXEC.BAT and executes it.

Windows XP Boot Process

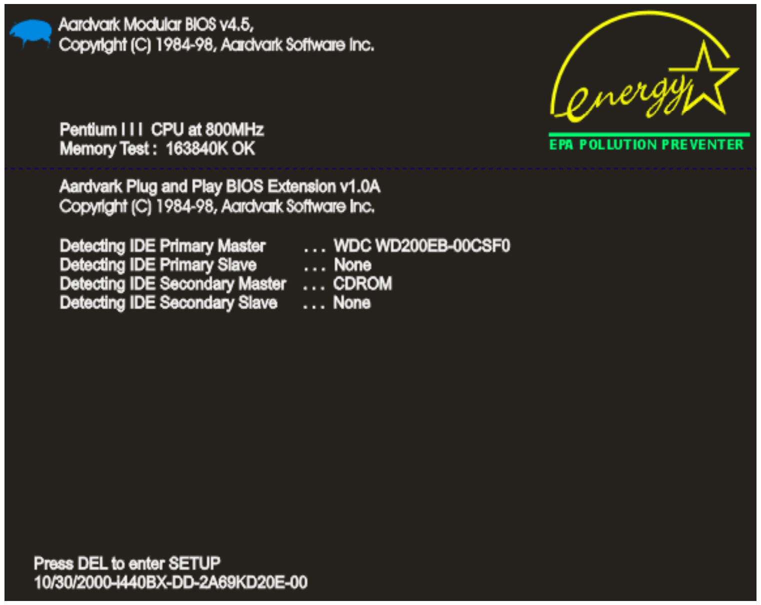
Windows XP Boot Process

After power supply is switched on:

- Power supply performs a self-test and sends the Power Good signal to the processor.
- Timer chip stops sending reset signals to the processor allowing the CPU to begin operations.
- CPU loads the ROM BIOS starting at ROM memory address FFFF:0000.
- ROM BIOS contains a JMP (jump) instruction that points to the actual address of the ROM BIOS code.
- ROM BIOS performs a basic test of central hardware to verify basic functionality.

Windows XP Boot Process (cont’d)

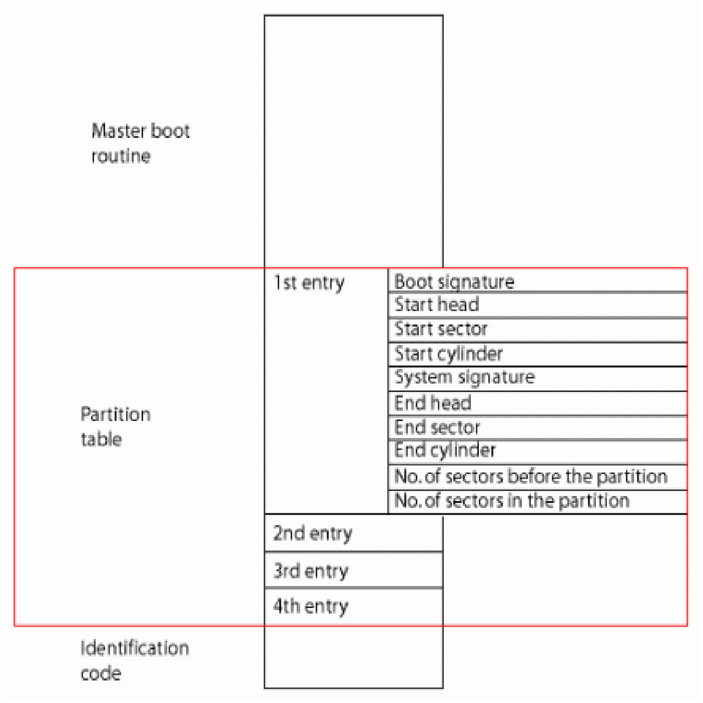
- The BIOS searches for adapters that may need to load their own ROM BIOS routines.
- Start-up BIOS routines scan memory addresses C000:0000 through C780:0000 to find video ROM.
- ROM BIOS checks to see if this is a 'cold-start' or a 'warm-start‘.
- If this is a cold-start the ROM BIOS executes a full POST (Power On Self Test). If this is a warm-start the memory test portion of the POST is switched off.
- POST can be broken down into two components:
  - Video Test initializes the video adapter.
  - Video adapter tests the video card and video memory and displays configuration information or any errors.



```
Aardvark Modular BIOS v4.5,
Copyright (C) 1984-98, Aardvark Software Inc.


Pentium I I I  CPU at 800MHz
Memory Test :  163840K OK

Aardvark Plug and Play BIOS Extension v1.0A
Copyright (C) 1984-98, Aardvark Software Inc.

Detecting IDE Primary Master      . . .  WDC WD200EB-00CSF0
Detecting IDE Primary Slave       . . .  None
Detecting IDE Secondary Master    . . .  CDROM
Detecting IDE Secondary Slave     . . .  None


Press DEL to enter SETUP
10/30/2000-I440BX-DD-2A69KD20E-00
```

Windows XP Boot Process (cont'd)

- BIOS locates and reads the configuration information stored in CMOS.
- BIOS examines the disk for a Master Boot Record (MBR).
- With a valid MBR loaded into memory, BIOS transfers control of the boot process to the partition loader code.
- Partition loader or boot loader examines the partition table for a partition marked as active.
- Partition loader then searches the very first sector of that partition for a boot record.
- The active partition's boot record is checked for a valid boot signature and if found the boot sector code is executed as a program.

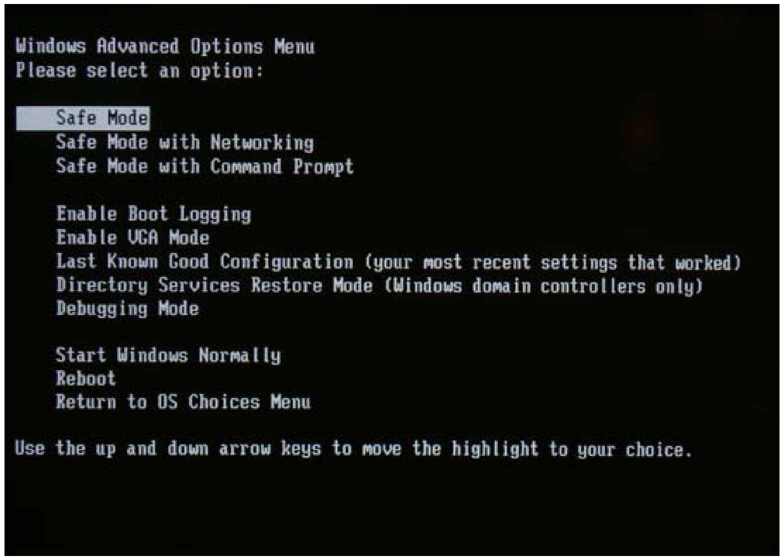


Windows XP Boot Process (cont’d)

- NTLDR, a hidden system file in the root directory of the system partition, controls loading of Windows XP in four stages:
  - Initial Boot Loader Phase
    - NTLDR switches the processor from real-mode to protected mode which places the processor in 32-bit memory mode and turns memory paging on.
    - It then loads the appropriate mini-file system drivers to allow NTLDR to load files from a partition formatted with any of the files systems supported by XP.
    - If the file BOOT.INI is located in the root directory NTLDR will read it's contents into memory.

Windows XP Boot Process (cont’d)

- Operating System selection
  - If BOOT.INI contains entries for more than one operating system NTLDR will stop the boot sequence at this point, display a menu of choices, and wait for a specified period of time for the user to make a selection.
  - Pressing F8 at this stage of the boot sequence to display various boot options including "Safe Mode" and "Last Known Good Configuration” (in Windows NT, 2000, or XP).



```
Windows Advanced Options Menu
Please select an option:

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable VGA Mode
    Last Known Good Configuration (your most recent settings that worked)
    Directory Services Restore Mode (Windows domain controllers only)
    Debugging Mode

    Start Windows Normally
    Reboot
    Return to OS Choices Menu

Use the up and down arrow keys to move the highlight to your choice.
```

EC-Council

Windows XP Boot Process (cont'd)

- **Hardware Detection**
  - If the selected operating system is XP, NTLDR locates and loads the DOS based NTDETECT.COM program to perform hardware detection.
  - If this computer has more than one defined Hardware Profile, NTLDR will stop at this point and display the Hardware Profiles/Configuration Recovery menu.
  - After selecting a hardware configuration NTLDR begins loading the XP kernel (NTOSKRNL.EXE).

Windows XP Boot Process (cont'd)

- Configuration Selection
  - NTLDR now loads device drivers that are marked as boot devices. With the loading of these drivers NTLDR relinquishes control of the computer.
  - NTOSKRNL goes through two phases in its boot process.

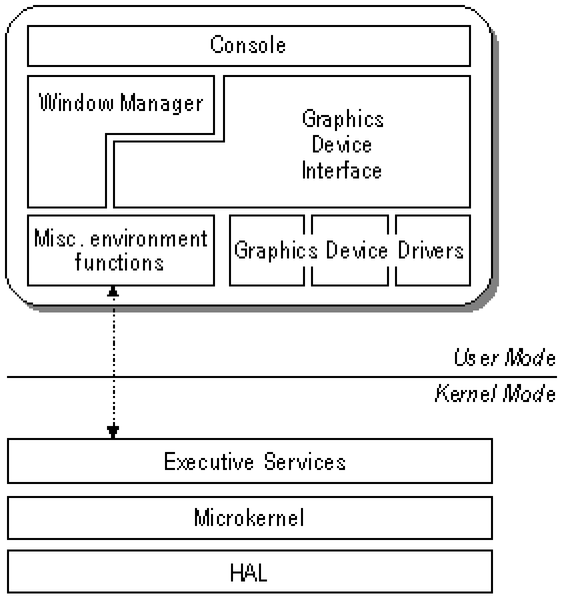
    phase 0:
    - XP disables interrupts during phase 0 and enables them before phase 1.
    - The HAL is called to prepare the interrupt controller.

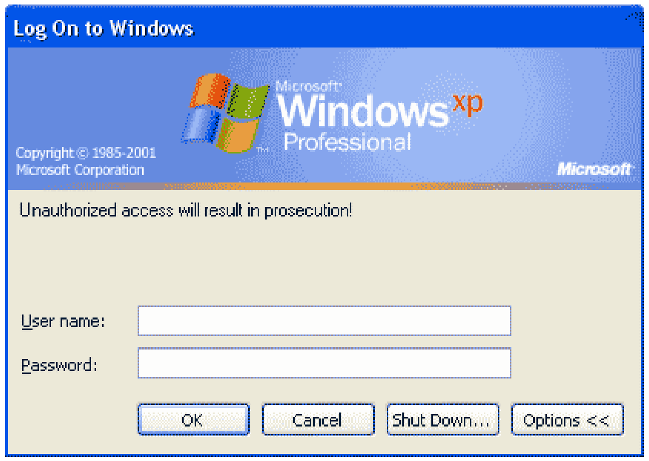
    phase 1:
    - All Executive subsystems are reinitialized in the following order:
      1. Object Manager
      2. Executive
      3. Microkernel
      4. Security Reference Monitor
      5. Memory Manager
      6. Cache Manager
      7. LPCS
      8. I/O Manager
      9. Process Manager

EC-Council

Windows XP Boot Process (cont'd)

- I/O Manager starts loading all the systems driver files.
- it first finishes the loading of boot devices.
- it assembles a prioritized list of drivers and attempts to load each in turn.
- launches the Session Manager Subsystem (SMSS).
- SMSS loads the win32k.sys device driver which implements the Win32 graphics subsystem.
- win32k.sys switches the screen into graphics mode.
- Services Subsystem starts all services mark as Auto Start.
- Once all devices and services are started the boot is deemed successful and this configuration is saved as the Last Known Good Configuration.
- The logging process is started by the WINLOGON.EXE file.
- Local Security Authority (LSASS.EXE) process displays the logon dialog box.




Log On to Windows

Copyright © 1985-2001 Microsoft Corporation

Microsoft Windows xp Professional

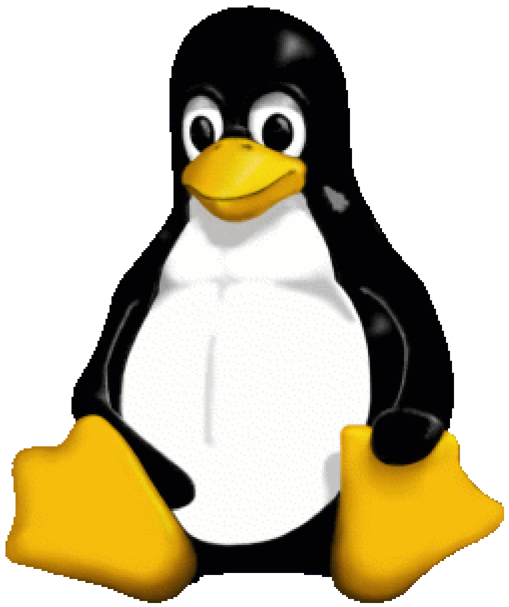
Unauthorized access will result in prosecution!

User name:

Password:

EC-Council

Linux Boot Process



Common Startup Files in UNIX

- .bashrc
  - For the bash shell. The file is a shell script, which means it can contain commands and other programming constructs.
- .bash_profile
  - For the bash shell. Another shell script. The difference between this script and *.bashrc* is that *.bash_profile* runs only when you log in. It was originally designed so you could separate interactive shells from those run by background processors like cron.
- .cshrc
  - For the C shell or tcsh. The file is a shell script using C shell construct.

Source: http://www.unix.org.ua/orelly/linux/run/ch04_15.htm

Common Startup Files in UNIX (cont'd)

- .login
  - For the C shell or tcsh. The file is a shell script, using C shell constructs. Like *.bash_profile* in the bash shell, this runs only when you log in.
- .emacs
  - For the Emacs editor. Consists of LISP function.
- *.exrc*
  - For the vi editor (also known as ex). Each line is an editor command.
- .fvwm2rc
  - For the fvwm2 window manager. Consists of special commands interpreted by fvwm2.

Source: http://www.unix.org.ua/orelly/linux/run/ch04_15.htm

Common Startup Files in UNIX (cont'd)

- .twmrc
  - For the twm window manager. Consists of special commands interpreted by twm.
- .newsrc
  - For news readers. Contains a list of all newsgroups offered at the site.
- .Xdefaults
  - For programs using the X Window System. Each line specifies a resource along with the value that resource should take.
- .xinitrc
  - For the X Window System. Consists of shell commands that run whenever you log into an X session.

Source: http://www.unix.org.ua/orelly/linux/run/ch04_15.htm

List of Important Directories in UNIX

- /bin
  - The most essential Unix commands, such as ls.
- /usr/bin
  - Other commands. The distinction between /bin and /usr/bin is arbitrary; it was a convenient way to split up commands on early Unix systems that had small disks.
- /usr/sbin
  - Commands used by the superuser for system administration.
- /boot
  - Location where the kernel and other files used during booting are sometimes stored.
- /etc
  - Files used by subsystems such as networking, NFS, and mail. Typically, these contain tables of network services, disks to mount, and so on.
- /var
  - Administrative files, such as log files, used by various utilities.
- /var/spool
  - Temporary storage for files being printed, sent by UUCP, and so on.

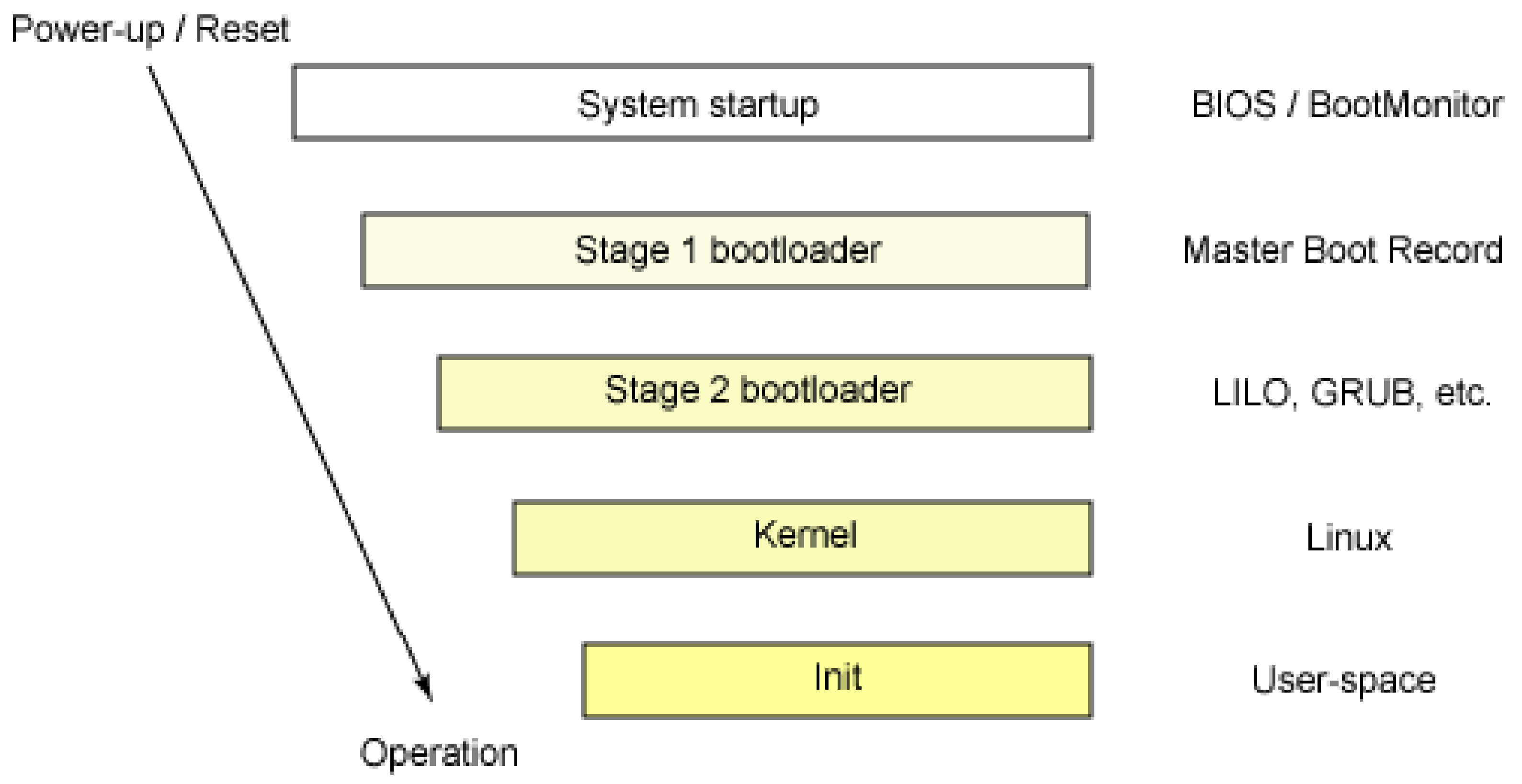
Source: http://www.unix.org.ua/orelly/linux/run/ch04_16.htm

List of Important Directories in UNIX (cont'd)

- /usr/lib
  - Standard libraries, such as libc.a. When you link a program, the linker always searches here for the libraries specified in -l options.
- /usr/lib/X11
  - The X Window System distribution. Contains the libraries used by X clients, as well as fonts, sample resources files, and other important parts of the X package. This directory is usually a symbolic link to /usr/X11R6/lib/X11.
- /usr/include
  - Standard location of include files used in C programs, such as <stdio.h>.
- /usr/src
  - Location of sources to programs built on the system.
- /usr/local
  - Programs and data files that have been added locally by the system administrator.
- /etc/skel
  - Sample startup files you can place in home directories for new users.

Source: http://www.unix.org.ua/orelly/linux/run/ch04_16.htm

Linux Boot Process



Source: http://www-128.ibm.com/developerworks/library/l-linuxboot/index.html

Linux Boot Process Steps

- Step 1: The Boot Manager
  - It is responsible for starting the booting process.
- Step 2: init
  - This process initializes the booting.
- Step 2.1: /etc/inittab
  - It provides the 4 fields as id, runlevels, action and process.
- Step 3: Services
  - It selects the services.
- Step 4: More inittab

Step 1: The Boot Manager

- Boot manager is a small program that resides mostly on the MBR.
- It displays a menu letting user to choose what operating system (if more than one OS) to boot.
- LILO is the common boot loader in Linux. It performs the following tasks:
  - Loads the kernel into the memory directly from the physical sector on a hard disk.
  - Optionally loads a ramdisk called initrd containing stuff like disk drivers.
  - Passes the kernel arguments like run level and init.
  - Starts execution of the kernel.

Step 1: The Boot Manager (cont'd)

- The kernel is responsible for recognizing and including all the hardware into the system.
- The last task of the kernel is to mount the root partition.
- The root partition is specified in *LILO* as a parameter. Then it starts the first process, which is usually called INIT.
- The messages that show the point where the kernel ends and the programs begin looks like this:
    - **VFS**: mounted root (ext2fs) filesystem read-only (still Kernel)
    - **INIT**: Version X.XX booting (already INIT)

Step 2: init

- Boot loader passes control to `/sbin/init.`
- Linux **`init`** will then read a file called `/etc/inittab.`
- Run level
  - A runlevel is a state for the system. Usually `0, 1, 2, 3, 4, 5, 6 and S.`
  - For example, pass `init=/bin/sh` to the kernel, and then a plain shell would be used.

Step 2: init (cont'd)

- Now, INIT checks which program has to be started next.
- It executes si:I:wait:PROGRAM. This may lead to any one or all of the following:
  - Startup of a server
  - Startup of shell
  - Network connection
  - Making the partition
- In general PROGRAM stands for /etc/rc.d/init.boot or /sbin/init.d/boot.

Step 2.1: /etc/inittab

```
1:2345:respawn:/sbin/mingetty tty1
```

- They have four fields, separated by colons.
- **id**
  - This has no real meaning, but should be different for each line, can be one to four characters.
- **runlevels**
  - For example, 2345 means this line applies to runlevels 2,3,4 and 5.
- **action**
  - Action of the line.
- **process**
  - A command to be executed.

runlevels

```
# The runlevels used by RHS are:

#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not
    have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
```

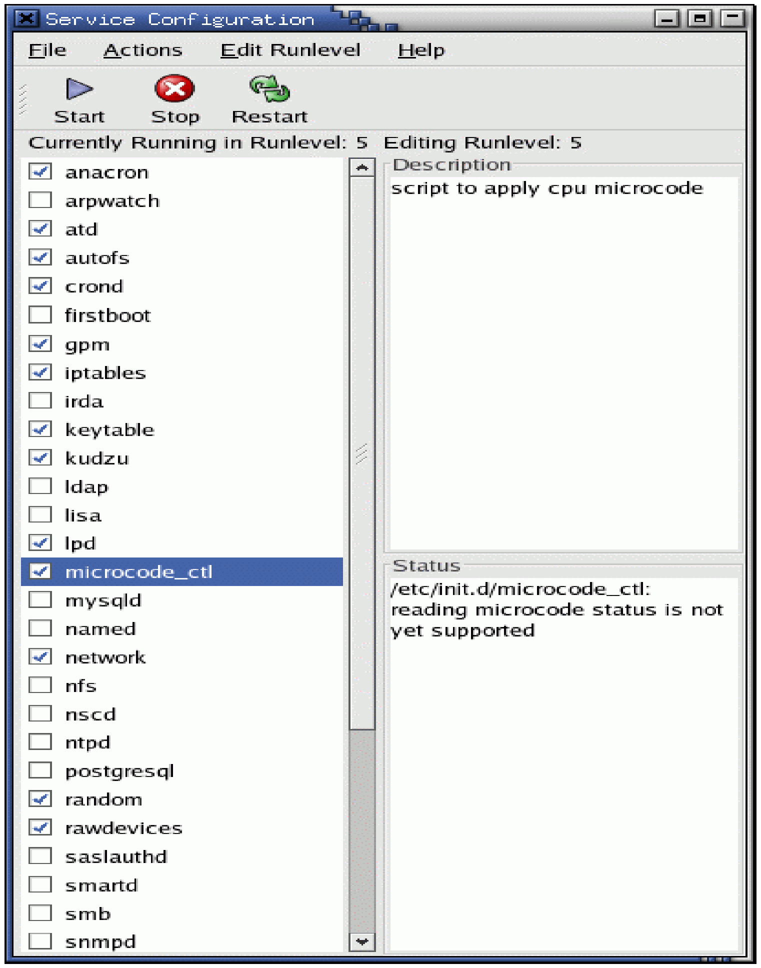
Step 3: Services

- Services are specified here

  `/etc/init.d/` or

  `/etc/rc.d/init.d`

- Some services support more or less commands, but all support stop, start, and restart.
- For each runlevel, there's a list of services that should be started, and a list of services that should be stopped.



Currently Running in Runlevel: 5   Editing Runlevel: 5

- [✓] anacron
- [ ] arpwatch
- [✓] atd
- [✓] autofs
- [✓] crond
- [ ] firstboot
- [✓] gpm
- [✓] iptables
- [ ] irda
- [✓] keytable
- [✓] kudzu
- [ ] ldap
- [ ] lisa
- [✓] lpd
- [✓] microcode_ctl
- [ ] mysqld
- [ ] named
- [✓] network
- [ ] nfs
- [ ] nscd
- [ ] ntpd
- [ ] postgresql
- [✓] random
- [✓] rawdevices
- [ ] saslauthd
- [ ] smartd
- [ ] smb
- [ ] snmpd

Description

script to apply cpu microcode

Status

/etc/init.d/microcode_ctl: reading microcode status is not yet supported

EC-Council

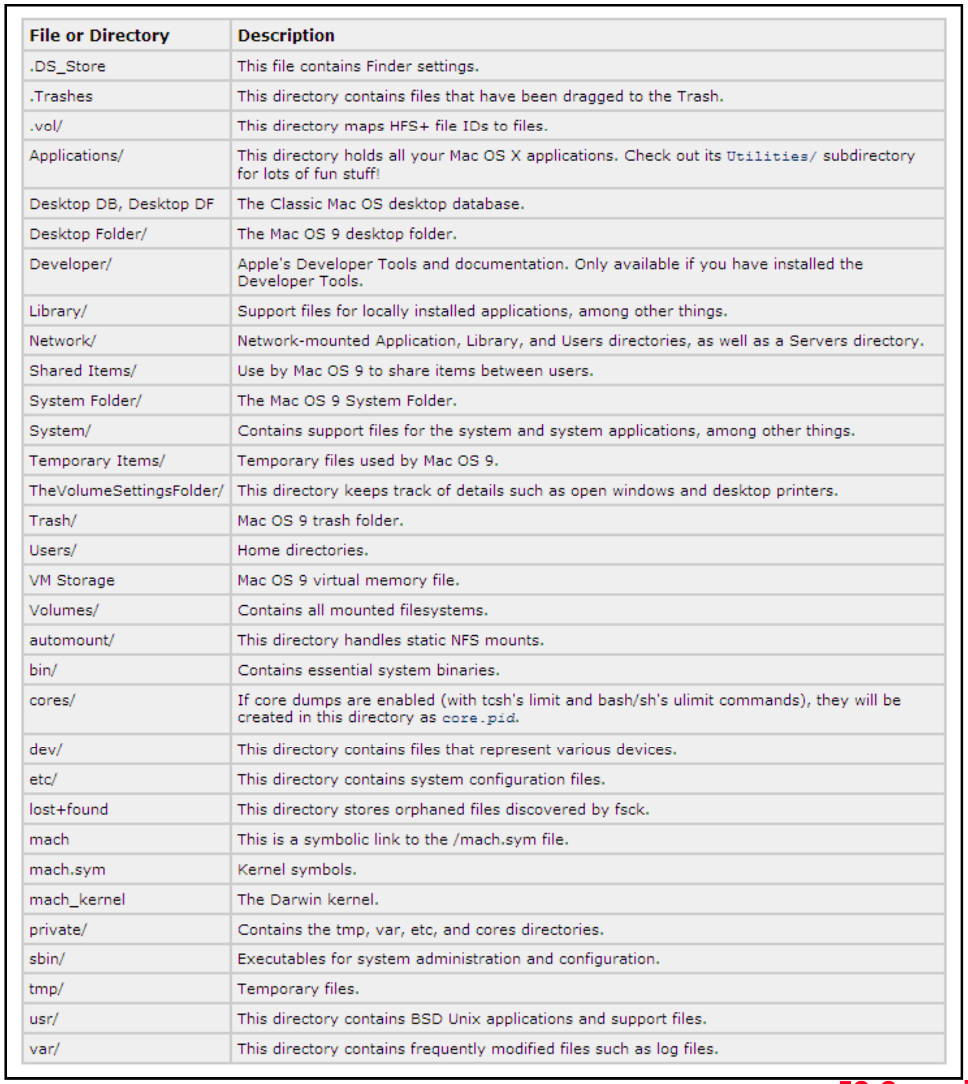
Step 4: More inittab

- Now **`init`** will get all lines with action respawn for the desired runlevel and start their processes.
- respawn commands are restarted when they end, so they will be running well all the time as long as you are in run level 5.
- Finally, system is booted and ready to login.

Macintosh Boot Process

Mac OS X

- Mac OS X is based on BSD Darwin engine.
- MAC OS X uses HFS+ file system.
- Mac OS X has nothing like the `/etc/init.d` directory. Instead, it finds its startup items in either `/System/Library/StartupItems` (for system startup items) or `/Library/StartupItems` (for locally-installed startup items).



File or Directory	Description
.DS_Store	This file contains Finder settings.
.Trashes	This directory contains files that have been dragged to the Trash.
.vol/	This directory maps HFS+ file IDs to files.
Applications/	This directory holds all your Mac OS X applications. Check out its `Utilities/` subdirectory for lots of fun stuff!
Desktop DB, Desktop DF	The Classic Mac OS desktop database.
Desktop Folder/	The Mac OS 9 desktop folder.
Developer/	Apple's Developer Tools and documentation. Only available if you have installed the Developer Tools.
Library/	Support files for locally installed applications, among other things.
Network/	Network-mounted Application, Library, and Users directories, as well as a Servers directory.
Shared Items/	Use by Mac OS 9 to share items between users.
System Folder/	The Mac OS 9 System Folder.
System/	Contains support files for the system and system applications, among other things.
Temporary Items/	Temporary files used by Mac OS 9.
TheVolumeSettingsFolder/	This directory keeps track of details such as open windows and desktop printers.
Trash/	Mac OS 9 trash folder.
Users/	Home directories.
VM Storage	Mac OS 9 virtual memory file.
Volumes/	Contains all mounted filesystems.
automount/	This directory handles static NFS mounts.
bin/	Contains essential system binaries.
cores/	If core dumps are enabled (with tcsh's limit and bash/sh's ulimit commands), they will be created in this directory as `core.pid`.
dev/	This directory contains files that represent various devices.
etc/	This directory contains system configuration files.
lost+found	This directory stores orphaned files discovered by fsck.
mach	This is a symbolic link to the /mach.sym file.
mach.sym	Kernel symbols.
mach_kernel	The Darwin kernel.
private/	Contains the tmp, var, etc, and cores directories.
sbin/	Executables for system administration and configuration.
tmp/	Temporary files.
usr/	This directory contains BSD Unix applications and support files.
var/	This directory contains frequently modified files such as log files.

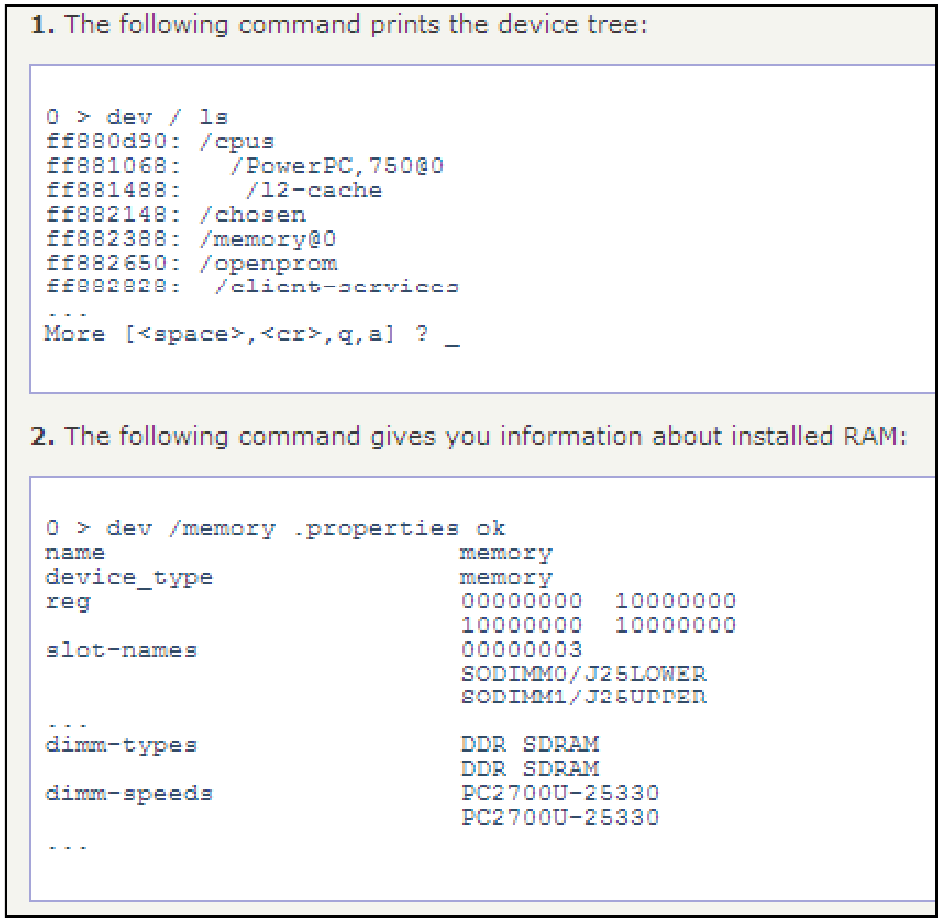
Mac OS X Hidden Files

- In Unix, a file can be made invisible by prefixing its name with a ., as in /.vol.
- HFS+ (a file system used by Mac OS) files and directories have a hidden attributes that can be set using SetFile command.
- `SetFile -a V SomeFile`

Booting Mac OS X

(Supported on non-Intel Macs)

- Booting in Mac OS X depends on three steps:
  - Mac's Open Firmware
  - Bootloader
  - Boot up sequence
- Open Firmware can be started by pressing cmd-opt-O-F.
- A boot loader can load kernels from various file systems.



1. The following command prints the device tree:

```
0 > dev / ls
ff880d90: /cpus
ff881068:   /PowerPC,750@0
ff881488:    /l2-cache
ff882148: /chosen
ff882388: /memory@0
ff882650: /openprom
ff882828:  /client-services
...
More [<space>,<cr>,q,a] ? _
```

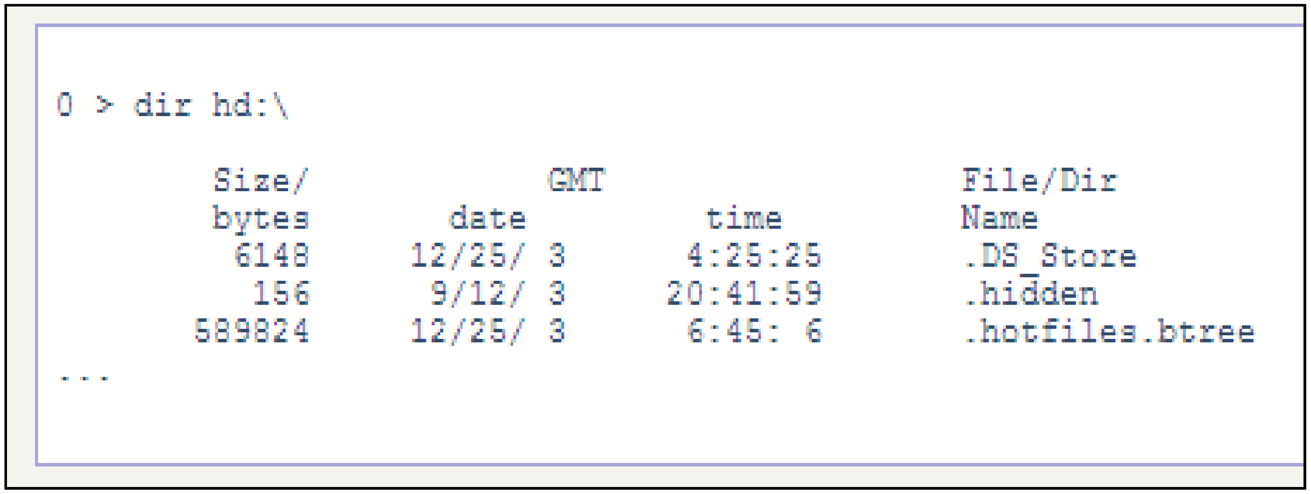
2. The following command gives you information about installed RAM:

```
0 > dev /memory .properties ok
name                       memory
device_type                memory
reg                        00000000  10000000
                           10000000  10000000
slot-names                 00000003
                           SODIMM0/J25LOWER
                           SODIMM1/J25UPPER
...
dimm-types                 DDR SDRAM
                           DDR SDRAM
dimm-speeds                PC2700U-25330
                           PC2700U-25330
...
```

EC-Council

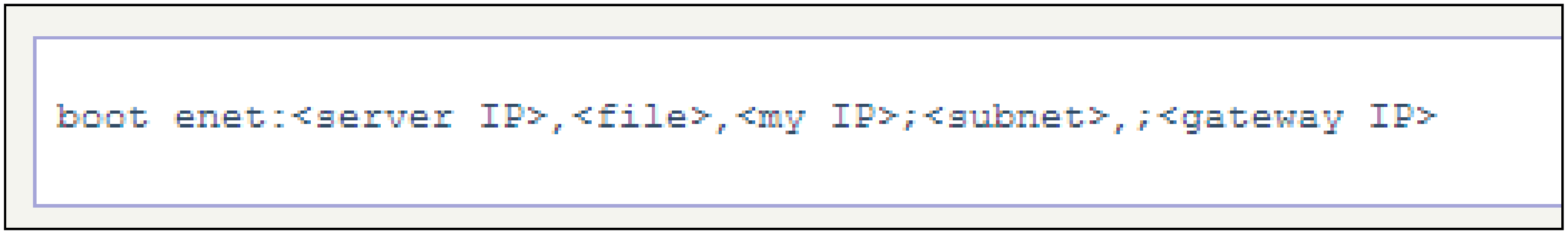
Screenshot

- Directory Listing



```
0 > dir hd:\

        Size/            GMT                   File/Dir
        bytes       date             time       Name
         6148      12/25/ 3       4:25:25      .DS_Store
          156       9/12/ 3      20:41:59      .hidden
       589824      12/25/ 3       6:45: 6      .hotfiles.btree
...
```

- Boot from TFTP Server



```
boot enet:<server IP>,<file>,<my IP>;<subnet>,;<gateway IP>
```

Mac OS X Boot Options

- `Command-S` Boot into Single User Mode.
- `Command-V` Boot using "Verbose" mode (shows all kernel and startup console messages).
- `X Reset` startup disk selection and boot into Mac OS X Server.
- Shift Boot into "Safe Boot" mode, which runs Disk First Aid. A reboot will be required afterward.
- `Option Boot` into Open Firmware to select a boot device.
- `Command-Option-Shift-Delete` Bypass internal harddrive on boot.
- `T Boot` into Firewire target disk mode.
- `C Boot` from the internal optical drive.
- `N Start` from the Network (NetBoot).
- `Command-Option-P-R` Reset Parameter RAM (PRAM) and non-volatile RAM (NVRAM).
- (mouse button) Eject (internal) removable media.

The Mac OS X Boot Process

- Mac OS X uses boot loader to perform the boot process.
- The Mac OS X Boot Process involves various steps:

1. Starts the OpenFirmware which Looks for a boot device.
2. OpenFirmware loads ’tbxi’ (BootX) file from partition.
3. It executes BootX which,
   - Reads root partition from nvram.
   - Loads mach kernel from the device.
   - Copies Mac OS X device drivers from partition into memory.
   - Disables all address translations.
   - Starts Mac OS X mach kernel.
4. Mach kernel begins its boot process.
5. Mac OS X desktop is loaded.

Installing Mac OS X on Windows XP

- Set path for prom_driver_graphic.
- Start and click install.
- Can customize the installation.
- Virtual machines like PearPC is used to install Mac OS X on Windows XP.
- PearPC is a way to get the OS X running in very less time span.

Source: http://pearpc.sourceforge.net

PearPC

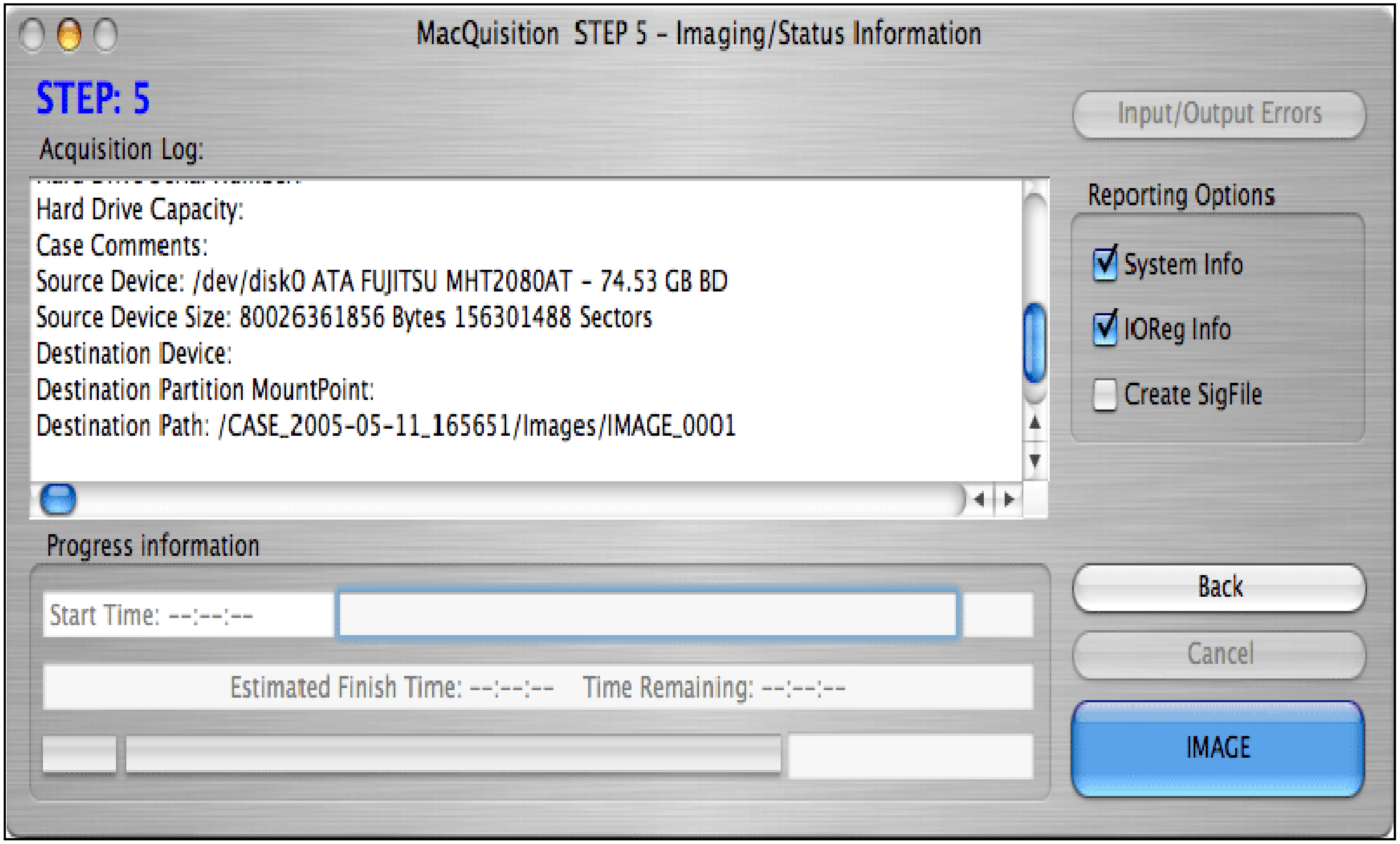
- ⊙ Capable of running most PowerPC OS.
- ⊙Behave as a client on certain operating systems.
- ⊙Limitations:
  - Performs well on small architectures.
  - Inaccurate timings.

MacQuisition Boot CD by BlackBag

- MacQuisition forensic acquisition tool utilized to image Mac's suspect drives using the suspect's own system safely and easily.
- Features:
  - Identify the suspected device(s).
  - Configure the image of destination location directly over the network.
  - Can use Command line.
  - Log case, exhibit and evidence tracking numbers and notes.
  - Generate MD5 hashes automatically.
  - Hash and block size customization with extension naming.

Source: http://www.blackbagtech.com

MacQuisition



Acquisition Log:

```
Hard Drive Capacity:
Case Comments:
Source Device: /dev/disk0 ATA FUJITSU MHT2080AT - 74.53 GB BD
Source Device Size: 80026361856 Bytes 156301488 Sectors
Destination Device:
Destination Partition MountPoint:
Destination Path: /CASE_2005-05-11_165651/Images/IMAGE_0001
```

Reporting Options

- [✓] System Info
- [✓] IOReg Info
- [ ] Create SigFile

Progress information

Start Time: --:--:--

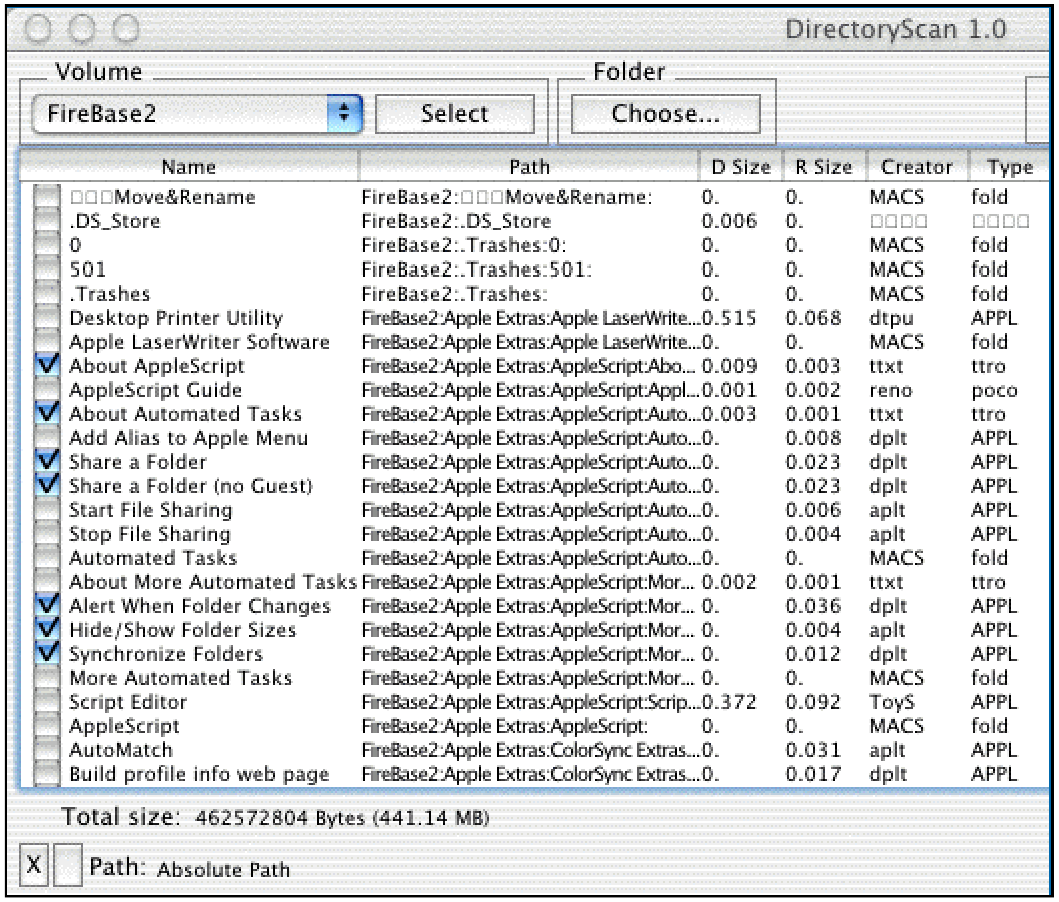
Estimated Finish Time: --:--:-- Time Remaining: --:--:--

Macintosh Forensic Software by BlackBag

- The BlackBag Macintosh Forensic Software is a unique set of 19 tools that provide forensic examiners with a flexible, open environment within which to perform their analysis.
- The Suite works within Mac Classic environments 8.1 - 9.2 and OS X.
- The applications are designed to efficiently carve and copy the most pertinent sectors of a target hard drive speeding the examiners analysis time, while ensuring a thorough investigation of the drive.
- Major applications contained within the suite are:
  - Directory Scan.
  - FileSpy.
  - HeaderBuilder.

Directory Scan

- Directory scan enables you to create a directory listing of a volume or specific folder.
- This scan will quickly retrieve all active file information (including invisible files) from a mounted volume.



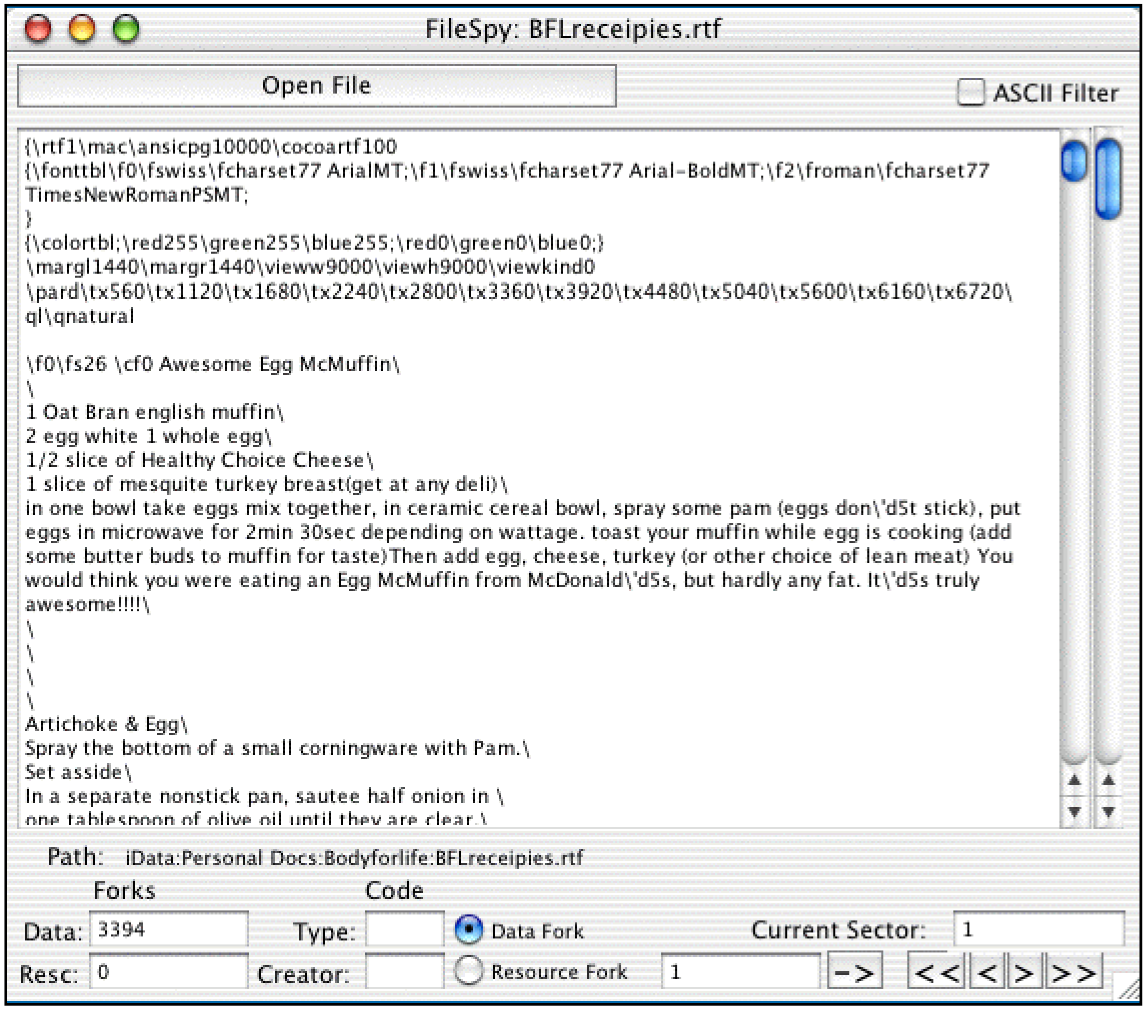
	Name	Path	D Size	R Size	Creator	Type
	□□□Move&Rename	FireBase2:□□□Move&Rename:	0.	0.	MACS	fold
	.DS_Store	FireBase2:.DS_Store	0.006	0.	□□□□	□□□□
	0	FireBase2:.Trashes:0:	0.	0.	MACS	fold
	501	FireBase2:.Trashes:501:	0.	0.	MACS	fold
	.Trashes	FireBase2:.Trashes:	0.	0.	MACS	fold
	Desktop Printer Utility	FireBase2:Apple Extras:Apple LaserWrite...	0.515	0.068	dtpu	APPL
	Apple LaserWriter Software	FireBase2:Apple Extras:Apple LaserWrite...	0.	0.	MACS	fold
✓	About AppleScript	FireBase2:Apple Extras:AppleScript:Abo...	0.009	0.003	ttxt	ttro
	AppleScript Guide	FireBase2:Apple Extras:AppleScript:Appl...	0.001	0.002	reno	poco
✓	About Automated Tasks	FireBase2:Apple Extras:AppleScript:Auto...	0.003	0.001	ttxt	ttro
	Add Alias to Apple Menu	FireBase2:Apple Extras:AppleScript:Auto...	0.	0.008	dplt	APPL
✓	Share a Folder	FireBase2:Apple Extras:AppleScript:Auto...	0.	0.023	dplt	APPL
✓	Share a Folder (no Guest)	FireBase2:Apple Extras:AppleScript:Auto...	0.	0.023	dplt	APPL
	Start File Sharing	FireBase2:Apple Extras:AppleScript:Auto...	0.	0.006	aplt	APPL
	Stop File Sharing	FireBase2:Apple Extras:AppleScript:Auto...	0.	0.004	aplt	APPL
	Automated Tasks	FireBase2:Apple Extras:AppleScript:Auto...	0.	0.	MACS	fold
	About More Automated Tasks	FireBase2:Apple Extras:AppleScript:Mor...	0.002	0.001	ttxt	ttro
✓	Alert When Folder Changes	FireBase2:Apple Extras:AppleScript:Mor...	0.	0.036	dplt	APPL
✓	Hide/Show Folder Sizes	FireBase2:Apple Extras:AppleScript:Mor...	0.	0.004	aplt	APPL
✓	Synchronize Folders	FireBase2:Apple Extras:AppleScript:Mor...	0.	0.012	dplt	APPL
	More Automated Tasks	FireBase2:Apple Extras:AppleScript:Mor...	0.	0.	MACS	fold
	Script Editor	FireBase2:Apple Extras:AppleScript:Scrip...	0.372	0.092	ToyS	APPL
	AppleScript	FireBase2:Apple Extras:AppleScript:	0.	0.	MACS	fold
	AutoMatch	FireBase2:Apple Extras:ColorSync Extras...	0.	0.031	aplt	APPL
	Build profile info web page	FireBase2:Apple Extras:ColorSync Extras...	0.	0.017	dplt	APPL

Total size: 462572804 Bytes (441.14 MB)

Path: Absolute Path

FileSpy

- FileSpy enables you to obtain a quick preview of any file by displaying the ASCII text for that file.
- You can move the file, sector by sector, jump to the start or end of a sector, or jump to any specific sector within the file.



```
{\rtf1\mac\ansicpg10000\cocoartf100
{\fonttbl\f0\fswiss\fcharset77 ArialMT;\f1\fswiss\fcharset77 Arial-BoldMT;\f2\froman\fcharset77
TimesNewRomanPSMT;
}
{\colortbl;\red255\green255\blue255;\red0\green0\blue0;}
\margl1440\margr1440\vieww9000\viewh9000\viewkind0
\pard\tx560\tx1120\tx1680\tx2240\tx2800\tx3360\tx3920\tx4480\tx5040\tx5600\tx6160\tx6720\
ql\qnatural

\f0\fs26 \cf0 Awesome Egg McMuffin\
\
1 Oat Bran english muffin\
2 egg white 1 whole egg\
1/2 slice of Healthy Choice Cheese\
1 slice of mesquite turkey breast(get at any deli)\
in one bowl take eggs mix together, in ceramic cereal bowl, spray some pam (eggs don\'d5t stick), put
eggs in microwave for 2min 30sec depending on wattage. toast your muffin while egg is cooking (add
some butter buds to muffin for taste)Then add egg, cheese, turkey (or other choice of lean meat) You
would think you were eating an Egg McMuffin from McDonald\'d5s, but hardly any fat. It\'d5s truly
awesome!!!!\
\
\
\
\
Artichoke & Egg\
Spray the bottom of a small corningware with Pam.\
Set asside\
In a separate nonstick pan, sautee half onion in \
one tablespoon of olive oil until they are clear.\
```

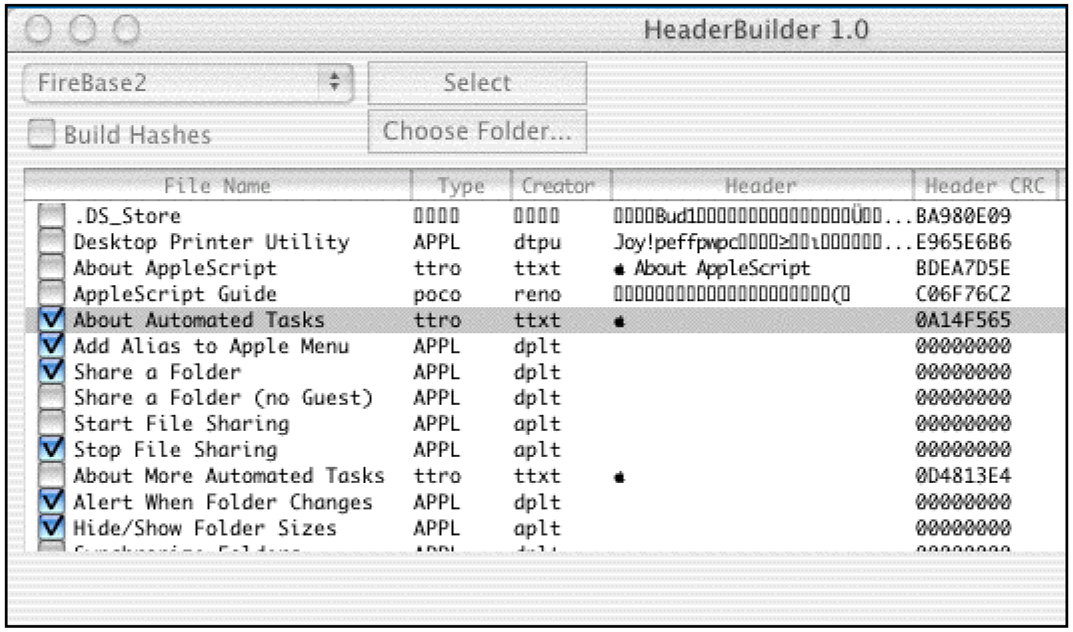
Path: iData:Personal Docs:Bodyforlife:BFLreceipies.rtf

Forks		Code			
Data:	3394	Type:		[✓ Data Fork]	Current Sector: 1
Resc:	0	Creator:		Resource Fork	1

EC-Council

HeaderBuilder

- HeaderBuilder enables you to build the header of specific files.
- It reads the first 32 bytes of each file.
- The header CRC will calculate a 32 bit CRC check sum of the first 32 bytes of the file and create an MD5 checksum of the entire file.

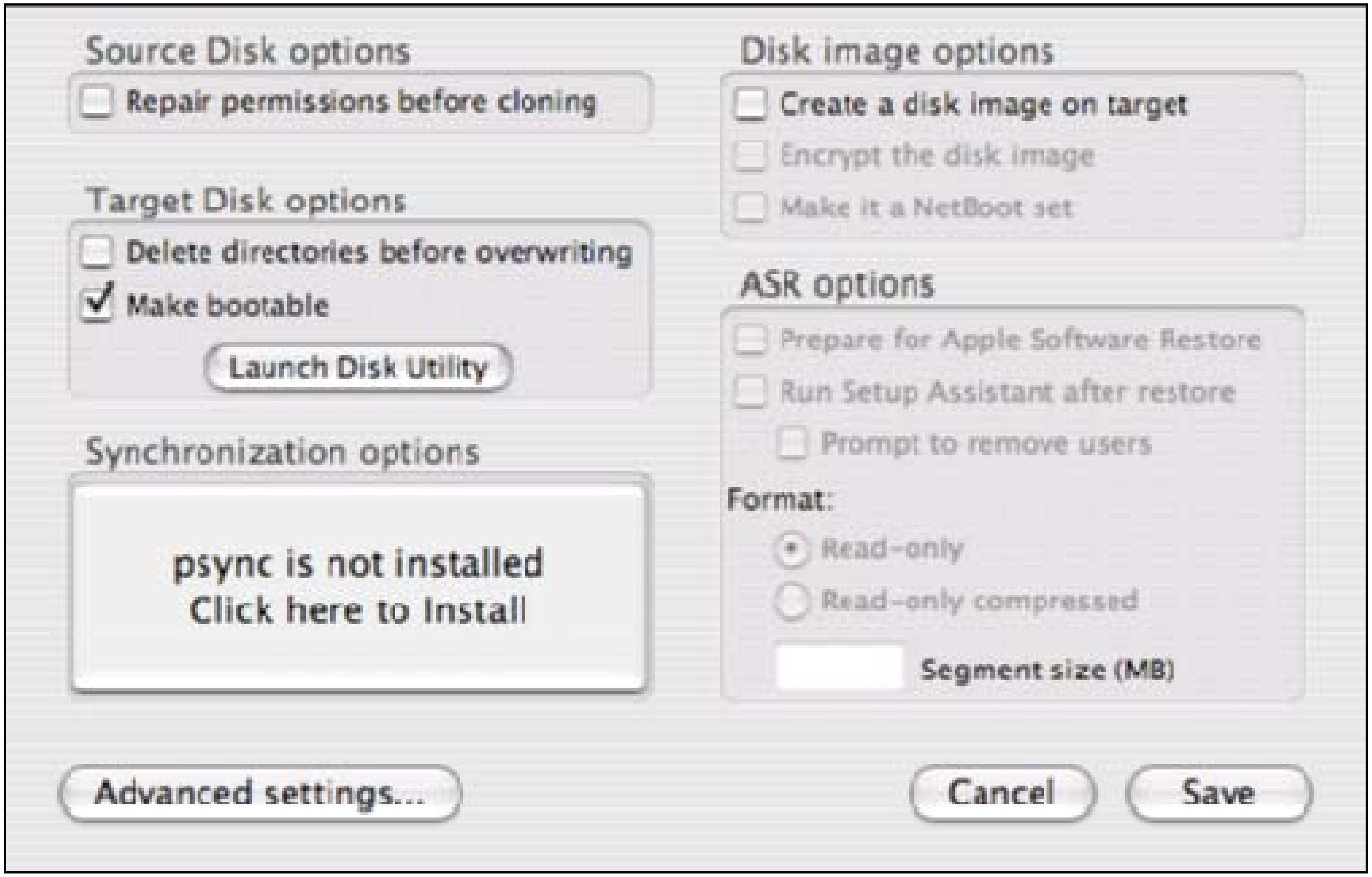


	File Name	Type	Creator	Header	Header CRC
	.DS_Store	0000	0000	0000Bud10000000000000000Ü00...	BA980E09
	Desktop Printer Utility	APPL	dtpu	Joy!peffpwpc0000≥00ı000000...	E965E6B6
	About AppleScript	ttro	ttxt	About AppleScript	BDEA7D5E
	AppleScript Guide	poco	reno	0000000000000000000000(0	C06F76C2
✓	About Automated Tasks	ttro	ttxt		0A14F565
✓	Add Alias to Apple Menu	APPL	dplt		00000000
✓	Share a Folder	APPL	dplt		00000000
	Share a Folder (no Guest)	APPL	dplt		00000000
	Start File Sharing	APPL	aplt		00000000
✓	Stop File Sharing	APPL	aplt		00000000
	About More Automated Tasks	ttro	ttxt		0D4813E4
✓	Alert When Folder Changes	APPL	dplt		00000000
✓	Hide/Show Folder Sizes	APPL	aplt		00000000

Carbon Copy Cloner (CCC)

- CCC makes the following tasks simple by harnessing the Unix power built into Mac OS X:
  - It is a simple, complete, bootable backup of hard drive.
  - It can upgrade to a larger hard drive with minimal hassle and without reinstalling OS and all of your applications.
  - It can move entire Mac OS X installation to a new computer.
  - It can setup a regular backup regimen that occurs in the background, even if you are not logged in.
  - It runs pre and post-flight shell scripts.
  - It can modify the list of items to be removed at the end of a clone.

Carbon Copy Cloner: Screenshot



Source Disk options

- [ ] Repair permissions before cloning

Target Disk options

- [ ] Delete directories before overwriting
- [✓] Make bootable

Synchronization options

psync is not installed
Click here to Install

Disk image options

- [ ] Create a disk image on target
- [ ] Encrypt the disk image
- [ ] Make it a NetBoot set

ASR options

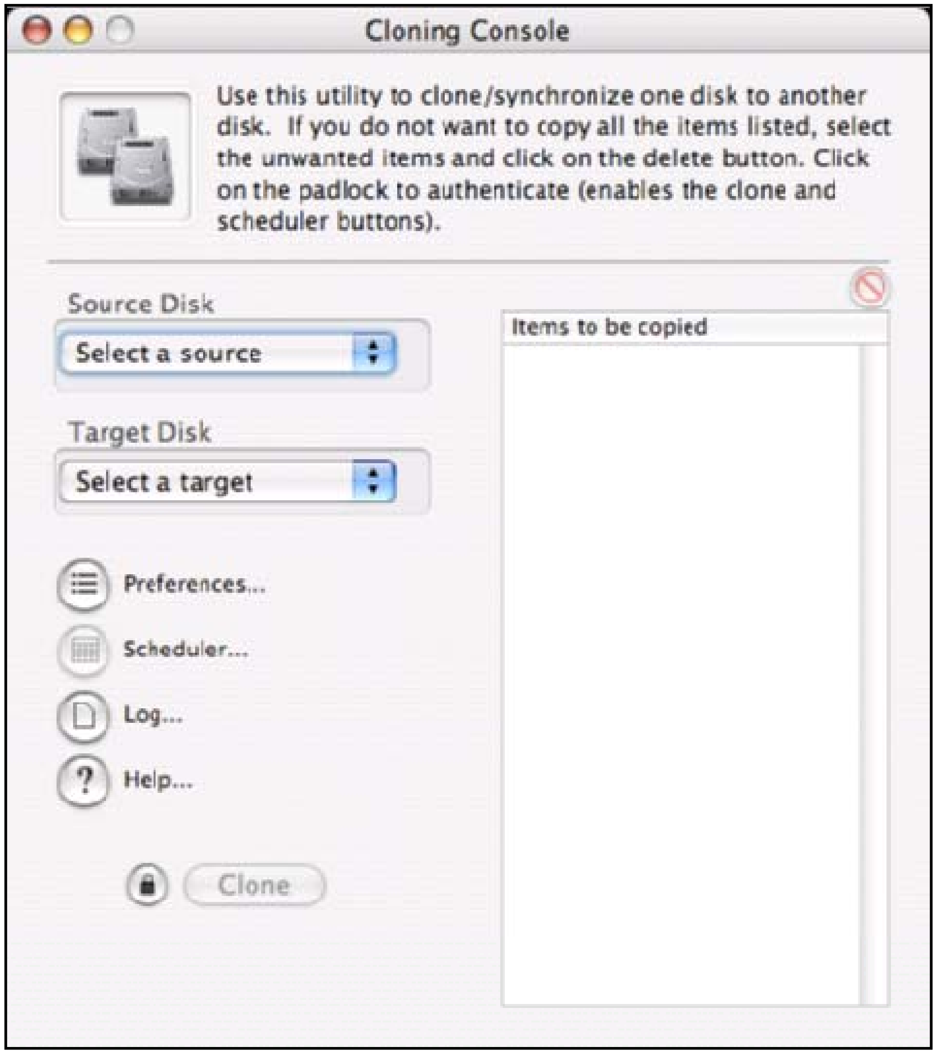
- [ ] Prepare for Apple Software Restore
- [ ] Run Setup Assistant after restore
- [ ] Prompt to remove users

Format: [● Read-only] Read-only compressed

Segment size (MB)

EC-Council

Carbon Copy Cloner: Screenshot

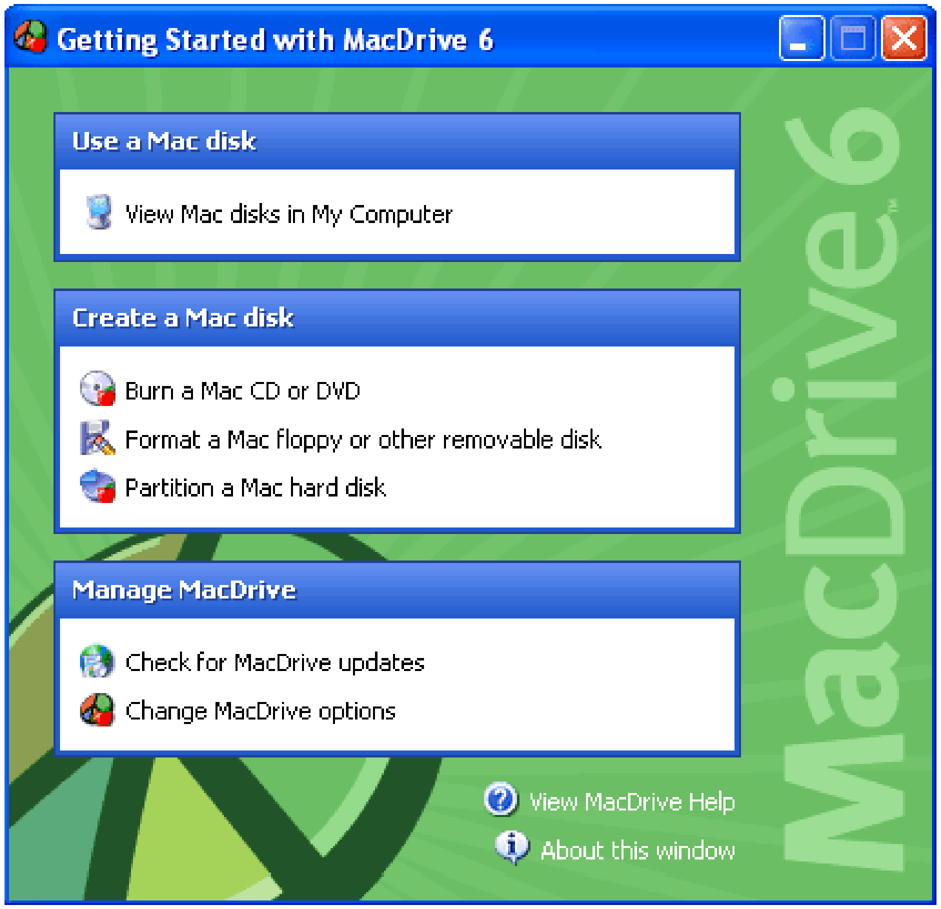


Use this utility to clone/synchronize one disk to another disk. If you do not want to copy all the items listed, select the unwanted items and click on the delete button. Click on the padlock to authenticate (enables the clone and scheduler buttons).

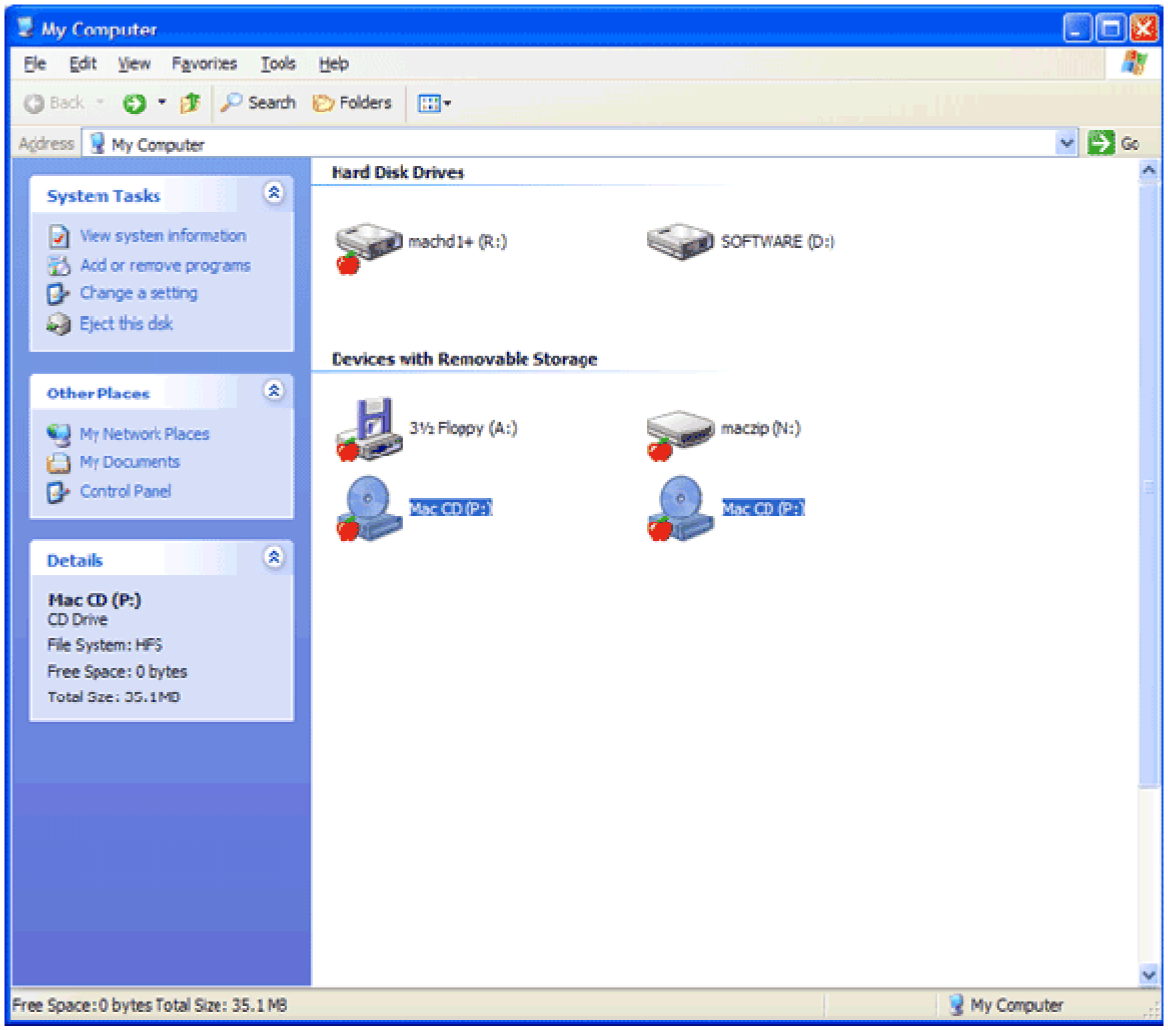
EC-Council

MacDrive6

- MacDrive is the ultimate solution for sharing files between Mac OS and Windows, perfect for graphic design, audio, video, education, digital photography, publishing, word processing, CAD/CAM, database administration.


Getting Started with MacDrive 6
Use a Mac disk
View Mac disks in My Computer
Create a Mac disk
Burn a Mac CD or DVD
Format a Mac floppy or other removable disk
Partition a Mac hard disk
Manage MacDrive
Check for MacDrive updates
Change MacDrive options
View MacDrive Help
About this window
MacDrive 6

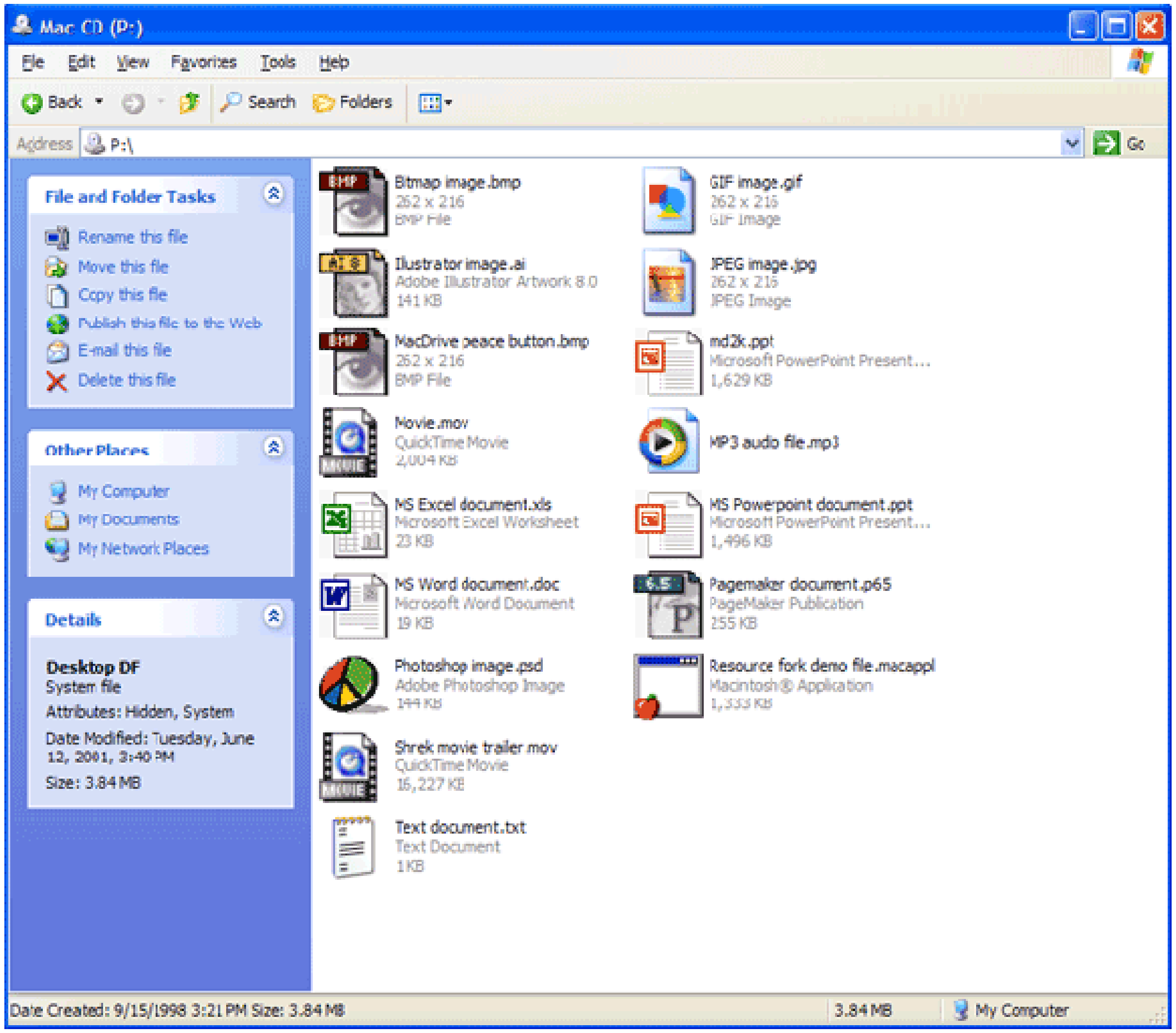
MacDrive6: Screenshot



Mac CD (P:)
CD Drive
File System: HFS
Free Space: 0 bytes
Total Size: 35.1 MB

Free Space: 0 bytes Total Size: 35.1 MB

MacDrive6: Screenshot



Desktop DF
System file
Attributes: Hidden, System
Date Modified: Tuesday, June 12, 2001, 3:40 PM
Size: 3.84 MB

Bitmap image.bmp
262 x 216
BMP File

GIF image.gif
262 x 216
GIF Image

Illustrator image.ai
Adobe Illustrator Artwork 8.0
141 KB

JPEG image.jpg
262 x 216
JPEG Image

MacDrive peace button.bmp
262 x 216
BMP File

md2k.ppt
Microsoft PowerPoint Present...
1,629 KB

Movie.mov
QuickTime Movie
2,004 KB

MP3 audio file.mp3

MS Excel document.xls
Microsoft Excel Worksheet
23 KB

MS Powerpoint document.ppt
Microsoft PowerPoint Present...
1,496 KB

MS Word document.doc
Microsoft Word Document
19 KB

Pagemaker document.p65
PageMaker Publication
255 KB

Photoshop image.psd
Adobe Photoshop Image
144 KB

Resource fork demo file.macappl
Macintosh® Application
1,333 KB

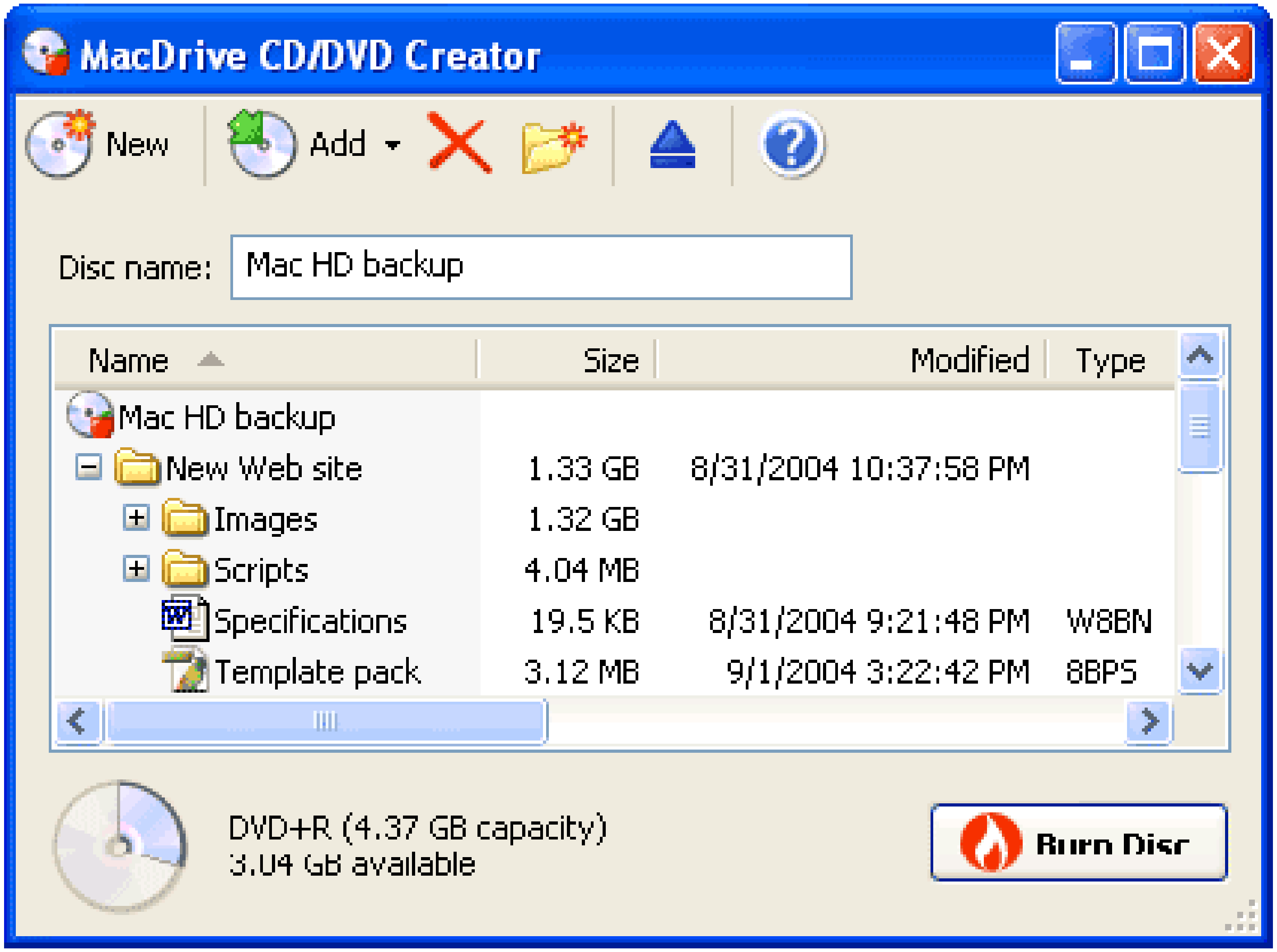
Shrek movie trailer.mov
QuickTime Movie
16,227 KB

Text document.txt
Text Document
1 KB

Date Created: 9/15/1998 3:21 PM Size: 3.84 MB 3.84 MB My Computer

EC-Council

MacDrive6: Screenshot



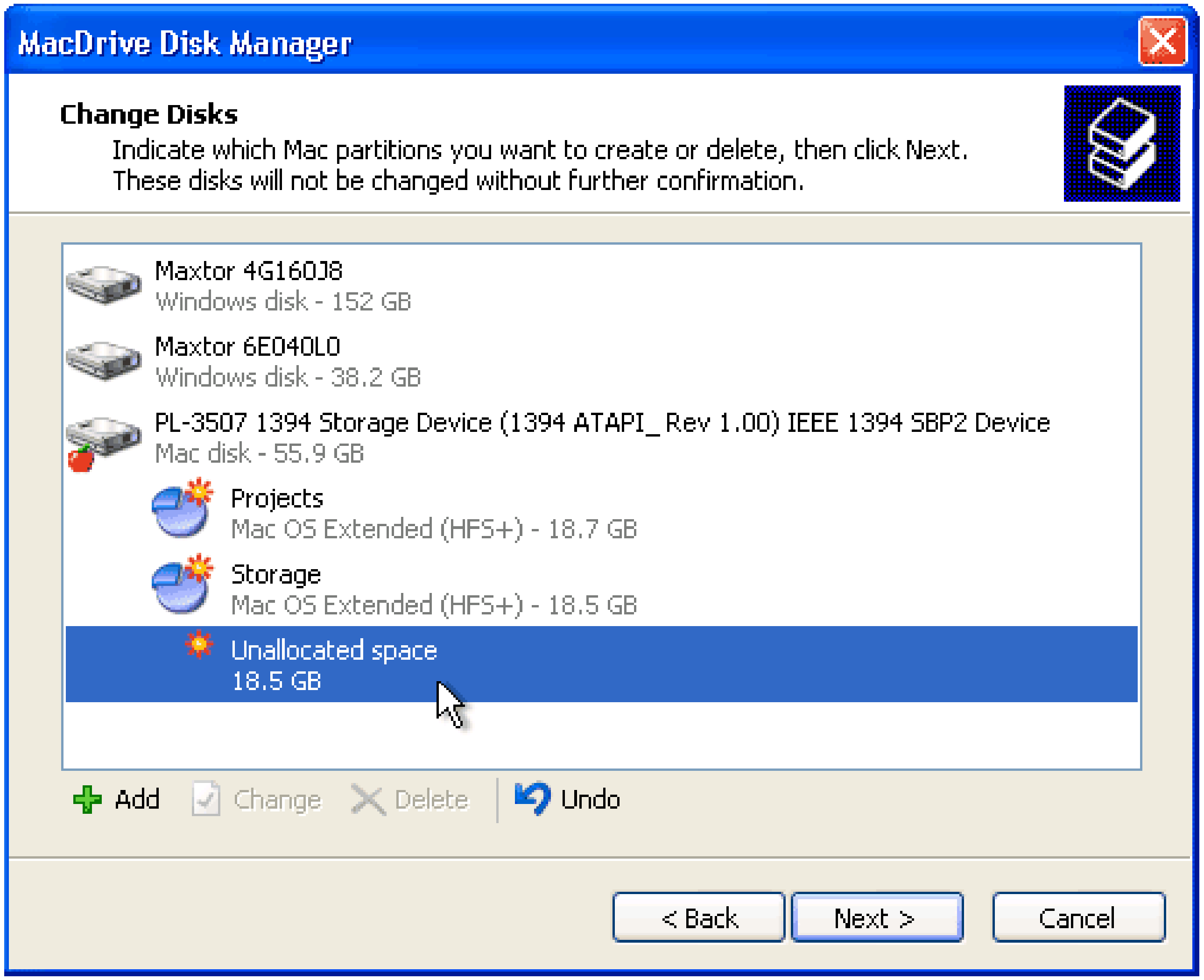
Disc name: Mac HD backup

Name	Size	Modified	Type
Mac HD backup			
New Web site	1.33 GB	8/31/2004 10:37:58 PM	
Images	1.32 GB		
Scripts	4.04 MB		
Specifications	19.5 KB	8/31/2004 9:21:48 PM	W8BN
Template pack	3.12 MB	9/1/2004 3:22:42 PM	8BPS

DVD+R (4.37 GB capacity)
3.04 GB available

EC-Council

MacDrive6: Screenshot



MacDrive Disk Manager

Change Disks

Indicate which Mac partitions you want to create or delete, then click Next.
These disks will not be changed without further confirmation.

Maxtor 4G160J8
Windows disk - 152 GB

Maxtor 6E040L0
Windows disk - 38.2 GB

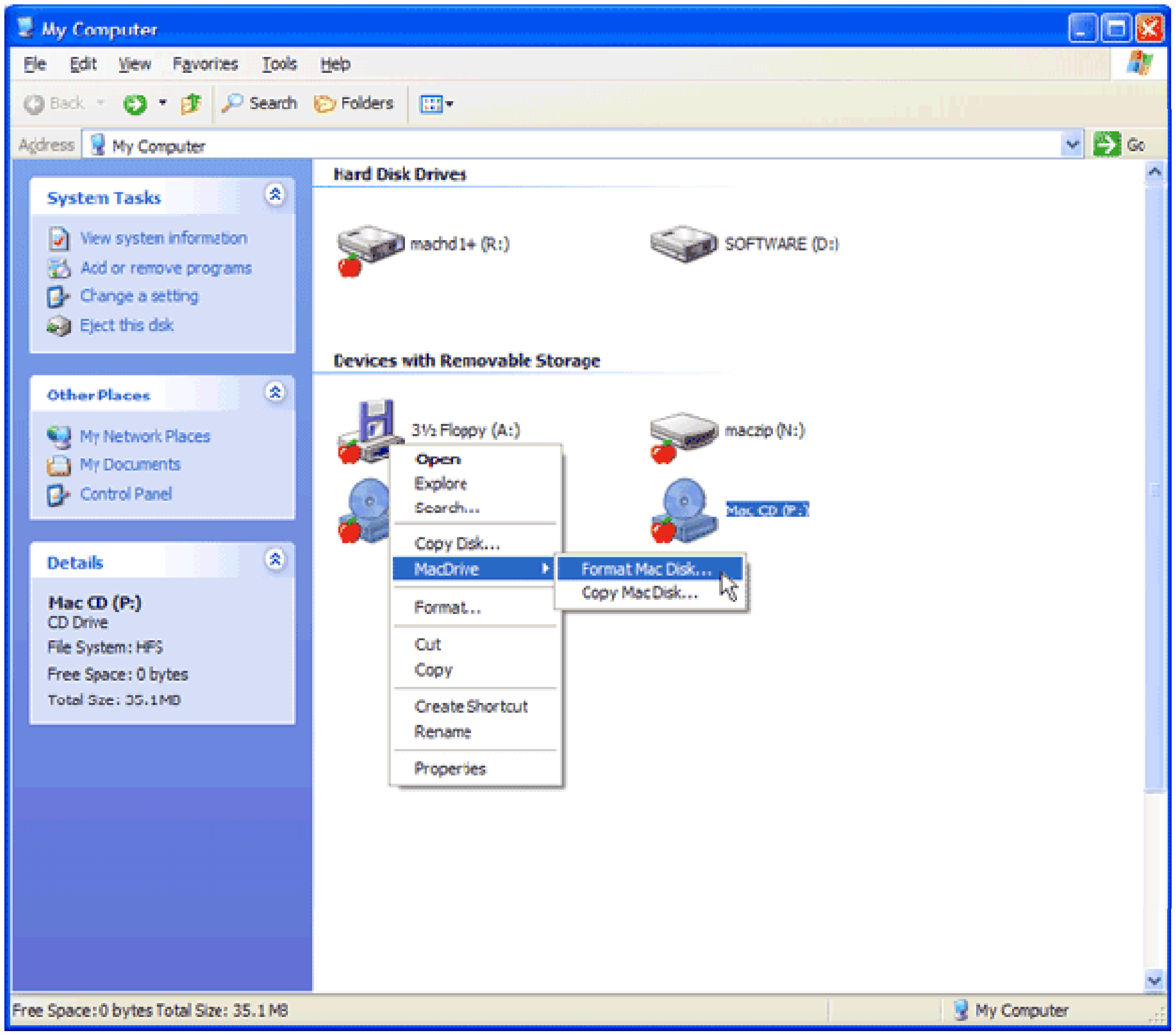
PL-3507 1394 Storage Device (1394 ATAPI_ Rev 1.00) IEEE 1394 SBP2 Device
Mac disk - 55.9 GB

Projects
Mac OS Extended (HFS+) - 18.7 GB

Storage
Mac OS Extended (HFS+) - 18.5 GB

Unallocated space
18.5 GB

MacDrive6: Screenshot



Mac CD (P:)
CD Drive
File System: HFS
Free Space: 0 bytes
Total Size: 35.1MB

Free Space: 0 bytes Total Size: 35.1 MB

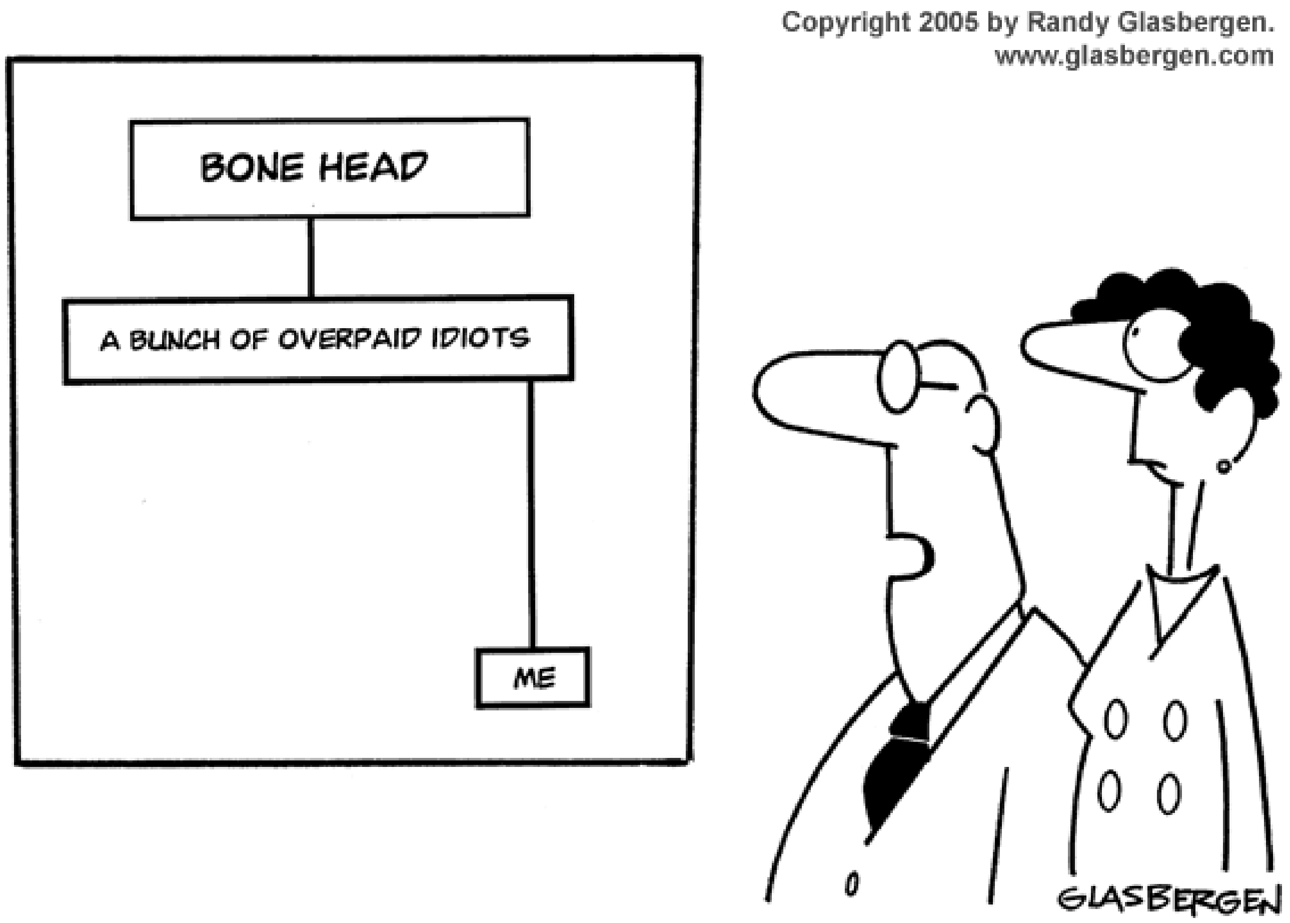
My Computer

EC-Council

Summary

- Booting is a process that starts operating systems when the user turns on a computer.
- Boot loader is a small program that loads the operating system into the computer's memory when the system is booted.
- Boot sector is a sector of a hard disk, floppy disk, or similar data storage device that contains code for bootstrapping programs.
- Master Boot Record is first sector of a data storage device that has been partitioned.
- Mac OS X is based on BSD Darwin engine.

"Every time you yell at me, your body temperature rises, the air conditioner has to work harder, energy costs go up, profits go down, and your stock loses value."


Copyright 2005 by Randy Glasbergen.
www.glasbergen.com
BONE HEAD
A BUNCH OF OVERPAID IDIOTS
ME
GLASBERGEN

"Who was put in charge of making the new organizational chart?"

“Somebody sent me a funny e-mail and I thought it was hilarious so I forwarded a copy to my boss. I didn’t know it was one of his memos!”