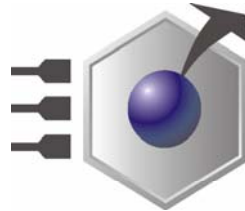


EmbeddedDNA[®]



An0071

**How to prepare a storage device to boot Windows CE
on x86 systems**

Rev. 1.1

Jan 2007

COPYRIGHT 1994-2007 Eurotech S.p.A. All Rights Reserved.



Via Fratelli Solari 3/a
33020 AMARO (UD)
ITALY

Phone: +39 0433 485 411
Fax: +39 0433 485 499

Website: www.eurotech.it
e-mail: sales@eurotech.it



NOTICE

Although all the information contained herein has been carefully verified, Eurotech S.p.A. assumes no responsibility for errors that might appear in this document, or for damage to property or persons resulting from an improper use of this manual and of the related software. Eurotech S.p.A. reserves the right to change the contents and form of this document, as well as the features and specifications of its products at any time, without notice.

Trademarks and registered trademarks appearing in this document are the property of their respective owners

Conventions

The following table lists conventions used throughout this guide.

Icon	Notice Type	Description
	Information note	Important features or instructions
	Warning	Information to alert you to potential damage to a program, system or device or potential personal injury

(This page is intentionally left blank.)

Contents

Conventions	3
Contents	5
Chapter 1 Introduction	7
Chapter 2 Setting up the BIOS bootloader	8
To use the floppy disk image	8
To prepare the storage device	8
Chapter 3 BIOS bootloader limitations.....	10
Where to find us	11

(This page is intentionally left blank.)

Chapter 1 Introduction

The x86 BIOS boot loader (biosloader) uses the BIOS services like the VESA BIOS for video display control, and INT 13h services for disk I/O to load an image from a local storage device.

It loads a .bin image from any device the BIOS exposes INT 13h support for and views as a storage device. This includes floppy, hard disk, DiskOnModule, Compact Flash (CF) on IDE , and Disk-On-Chip.

The boot loader resides on the bootable storage device and is found and loaded by a boot sector image.

The following list summarizes the BIOS boot loader boot process:

1. At power-on or reset, the x86 CPU jumps to the CPU reset vector address and starts to execute BIOS code.
The BIOS will typically configure the hardware, which includes configuring the DRAM controller, configuring the host bridge, and enumerating PCI devices, and then locating a boot drive.
2. If the boot drive is a fixed disk (for example, hard disk, Compact Flash card, or Disk-On-Chip) the first disk sector will be the master boot sector (MBR).
If the boot drive is a floppy disk, it does not contain an MBR. Instead, the first sector on the disk is a boot sector.
The MBR neatly fits in one disk sector and is made up of code, 16-bit real mode code, and data. The MBR is written to the disk by a partitioning tool like the MS-DOS fdisk program.
The MBR is loaded from the disk into RAM by the BIOS; then the BIOS begins to execute the MBR code.
If the BIOS cannot locate the MBR on the boot drive, which looks for a two-byte signature at the end of the sector, it displays an error message and halts.
3. The MBR code primarily searches through the partition table looking for the active partition on the boot disk.
The partition table is data that resides in the MBR.
If the MBR code cannot locate an active partition, it displays an error message and halts.
If it finds an active partition, it determines the disk address of that partition.
The first sector in the partition is called the boot sector. However, on a floppy disk, this is the first sector on the disk because there is no MBR on a floppy disk.
The MBR code loads the boot sector into RAM and jumps to it.
4. The boot sector, like the MBR, contains code and data and fits neatly within a single disk sector. The boot sector is the first piece of OS-specific code.
The boot sector's task is to find and load the BIOS boot loader. It does this by using the disk-resident properties of the native file system, along with BIOS services, to find and load the boot loader into RAM and then jump to it.

After starting, the BIOS loader proceeds.

To prepare a storage device for use with the BIOS boot loader, you must partition and format it.

For more information about the BIOS boot loader and its functionalities, see the following chapters.

Chapter 2 Setting up the BIOS bootloader

This application note refers to a compressed file that contains:

- Setupdisk.144 file
- Websetup.exe file
- Disk folder

Both setupdisk.144 and Websetup.exe are provided in Microsoft Platform Builder: Setupdisk.144 is an MS-DOS FAT-formatted floppy disk that contains disk setup utilities and the boot sector and boot loader images; Websetup.exe is an application which allow the creation of a floppy disk from setupdisk.144.

The compressed file can be downloaded from the Eurotech website (www.eurotech.it) and is located in the download area in the tools/utility section.

To use the floppy disk image

1. If Websetup is not installed on your system, install it by running Websetup.exe on your development workstation; Websetup only needs to be installed once.
2. Place a floppy disk in your floppy drive.
3. Run Setupdisk.144 (The .144 file extension is associated with Websetup).
4. Select the **A Drive** button and the floppy disk will be prepared.
5. Substitute *mkdisk.bat*, *bldr* and *boot.ini* files on the floppy disk with the homonymous files in the *Disk* folder

Note This step is necessary since the *bldr* file shipped with Setupdisk.144 file is affected by a bug.

To prepare the storage device

1. Insert the setup floppy disk and boot it.
2. From the command prompt, use the fdisk utility to create a primary MS-DOS active partition on the storage device.
Because the BIOS boot loader does not support FAT32, create a partition whose size is FAT12 or FAT16 compatible.
For more information about how to use fdisk, consult the fdisk documentation.

Note Prepare the disk in the target device. If you cannot, be sure the BIOS settings for the drive, such as the LBA mode and others, match the settings used on the hardware platform. On-disk settings are stored as part of the fdisk operation and must be compatible with the working BIOS mode.

3. From the command prompt, format the new partition using the format utility. The MS-DOS system files should not be transferred as part of the formatting process. Specifically, do not use *format /s <drive>*. When prompted by the format utility, leave the volume name blank.
4. Write the boot sector image and transfer the boot loader by typing **mkdisk <drive letter>:**. For example, if you are using the C: drive, you would type **mkdisk c:**.

At this point, the storage device contains an active partition that is bootable and contains the boot sector and boot loader images.

The mkdisk utility sets the attributes of the BIOS loader image (BLDR) to hide or protect the file.

Listing all files in the root directory of the storage device will not show files.

After the storage device is prepared with the BIOS loader, the BIN image to be booted must be transferred to the root directory.

Because the storage device contains a standard FAT file system, if the storage device is removable, it can be placed in any desktop system and the boot image can be copied to the root directory on the storage device.

When the BIOS loader boots, it sends information over physical COM1 on the target device. If you run into problems booting an image with the BIOS loader, connect a serial terminal (38400 8/N/1, no flow control) to COM1 for more information.

Chapter 3 BIOS bootloader limitations

- FAT32 is not supported.
- The boot loader assumes a VESA BIOS is available and the resolution is set to 640x480x16 by default. Display initialization code in the BIOS boot loader can be removed or modified to suit the hardware platform. By default, the BIOS boot loader tries to use the VESA BIOS to initialize the video.
- A flexible user configuration file (Boot.ini) is supported. Such a file allows the user to specify the image name to load, set video resolutions, and pass Ethernet debug NIC parameters.
- The BIOS boot loader does not contain decompression code to decompress a .bin file to save storage space.
- The boot loader does not provide a mechanism for a failsafe image boot in case the primary BIN file is corrupted.
- The storage device preparation utilities are MS-DOS utilities.

Where to find us

Eurotech S.p.A.

Via Jacopo Linussio, 1 - 33020 Amaro (UD) ITALY

Tel. +39 0433 485411

Fax +39 0433 485499

Welcome@eurotech.it

www.eurotech.it