



**Booting Windows CE From Cactus
Flash-Storage Products On x86 Platforms**

White Paper CTWP003

Cactus Technologies Limited

Suite C, 15/F, Capital Trade Center
62 Tsun Yip Street, Kwun Tong
Kowloon, Hong Kong
Tel: +852-2797-2277
Email: sales@cactus-tech.com

Cactus USA

3112 Windsor Road
Suite A356
Austin, Texas 78703
Tel: +512-775-0746
Email: americas@cactus-tech.com

Cactus-Tech.com

01**Introduction**

In many embedded applications, using traditional magnetic or optical storage devices proved to be difficult due to temperature, shock or vibration constraints. Using flash memory-based storage devices can be very helpful for the system to achieve high performance under these environmental constraints.

Microsoft Windows CE is a popular operating system (OS) for embedded applications and runs on multiple processor platforms. Many embedded platforms require the use of flash memory-based storage devices, such as CompactFlash cards or PCMCIA ATA cards. This application note outlines the procedures for preparing Cactus Technologies flash-storage (CF, PCMCIA, 2.5" SSD and DOM) devices to work with Microsoft Windows CE under the x86 CEPC platform.

02**The Windows CE Setup**

Windows CE provides the necessary tools to set up booting from an x86 CEPC platform using two methods:

- 1 MSDOS and `loadcepc.exe`.
- 2 Writing x86 BIOS boot loader to disk.

Procedures in this application note assume the Cactus device is used in True IDE mode, is visible under DOS as a fixed disk and a BIOS is available on the CEPC platform; also the procedures will work under both Windows CE 4.2 and 5.0. Any version related differences will be specifically mentioned.

03**Making the Setup Disk**

Windows CE includes the necessary tools to make the Cactus flash-memory storage device bootable using both methods above. Since these tools are MSDOS based, we need to create three floppy disks and use them at the target system (and NOT the development workstation) to prepare the Cactus flash storage device.

Before creating the setup diskette, the handler for ".144" files must be registered through the following one-time-only operation:

- Go to `%PROGRAMFILES%\Windows CE Platform Builder\<<VERSION>\CEPB\UTILITIES` where `<version>` is the Windows CE version and execute `websetup.exe`. This will install the utility Web Disk to generate disk images as used by Windows CE.
- Put a floppy in the A: drive and double click on `CEPCBoot.144` to generate the `CEPCBoot` floppy for booting Windows CE via `loadcepc.exe`.

- Go to %Wince% \PUBLIC\COMMON\OAK\CSP\X86\BIOS\LOADER\DISKIMAGES
- Put another floppy in the A: drive and double click SETUPDISK.144 . This is the SETUPDISK floppy containing the necessary tools to prepare the Cactus flash storage device for booting Windows CE.
- Put another floppy in the A: drive and double click BOOTDISK.144 . This is an example BOOTDISK floppy for the x86 BIOS boot loader.

04 Preparing the Cactus Product

The shipping configuration for Cactus flash storage device always contains a single formatted FAT/FAT32 partition covering the total device capacity which is compatible with most applications. However, this default partition information may not match the BIOS translation on the target system. Thus, it is necessary to recreate the partition table under the target system.

We use the following procedures to generate a matching partition table on the Cactus flash storage device.

- 1 Start the target system with the SETUPDISK floppy.
- 2 At the MSDOS command prompt, type `fdisk`. Consult `fdisk` documentation for detailed usage.
- 3 Delete the present C: partition by selecting 3 in the `fdisk` menu.
- 4 Then select 1 to create a partition, select 1 again for primary partition. Ensure that the Primary partition is less than 512MB. For devices with size less than 512MB , answer yes to use the full size; answer no for devices with sizes greater than 512MB. We have tested different partition sizes and found that Windows CE may not boot normally with primary partition size larger than 512MB.
- 5 Make the primary partition active by selecting 2. This creates the c: drive.
- 6 Create another extended partition for the remaining storage space. This becomes the D: drive.
- 7 Exit `fdisk`. The target system will automatically reboot.
- 8 At the MSDOS command prompt, type `format c:` to format the device.

05 Booting Via MS-DOS

This method enables the developer to quickly test out the Windows CE OS image without the need to debug the boot loader. This method uses MSDOS as a first-stage boot loader, then `loadcepc.exe` is used to start the Windows CE image after DOS is started.

The MSDOS boot sector and system files are required to be transferred to the device with this procedure. For MSDOS 6.22, this will consume around 250Kbyte of space on the device.

The following procedure prepares the Cactus device as an MSDOS bootable fixed disk and copies `loadcepc.exe` to the device. This procedure can be performed before or after the kernel image `nk.bin` is copied to the Cactus device.

- 1 Start the target system with an MSDOS 6.22 startup floppy. Microsoft Windows 95/98 startup floppy will also work. At the MSDOS command prompt, type `sys c:` to transfer MSDOS system files to the device.
- 2 Insert the CEPCBoot floppy and type the following commands to transfer `himem.sys` and `loadcepc.exe` to the device.

```
copy a:\loadcepc.exe c:\
copy a:\himem.sys c:\
```

- 3 On the development workstation, create the minimal MSDOS startup files `config.sys` and `autoexec.bat` as below. These files must be plain ASCII text files.

Config.sys

```
device=himem.sys /testmem:off
dos=high
```

Autoexec.bat

```
@echo off
loadcepc.exe nk.bin
```

- 4 Copy `config.sys` and `autoexec.bat` files to a floppy disk. Copy these files to the device on the target system.
- 5 Remove the floppy and reboot the target system. `Loadcepc` should display messages about PCI enumeration and other status before booting into Windows CE.

06

Booting Via BIOS Boot Loader

This method relies on the Windows CE boot loader code to locate and load the image from the Cactus device to memory. It only requires BIOS services on the target system / CEPC to start and no MSDOS system files are necessary.

To prepare the Cactus product for booting via BIOS boot loader, follow the procedures below for Windows CE .NET 4.2 :

- 1 Insert the `SETUPDISK` floppy created in Section 3, and under MSDOS command prompt, type `mkdisk c:` This command replaces the DOS boot sector on the Cactus device with the Windows CE boot sector and copies the Windows CE BIOS boot loader.
- 2 Remove the floppy disk and restart the target system.

07 Rebuilding & Booting Via BIOS Boot Loader (BLDR) For Windows CE 5.0

Windows CE 5.0 requires a different procedure for preparing the Cactus device for booting with the BIOS boot loader, as modification to the boot loader code is required to successfully boot with IDE-mode devices. The following procedure rebuilds the standard BIOS boot loader to accommodate booting Cactus devices.

- 1 Start the Windows CE 5.0 Command Prompt.

Go to `Public\common\oak\misc` and run the `winCE.bat` file with the target cpu type, project name and platform name:

```
wince.bat x86 cepctest CEPC
```

Note that the arguments are case sensitive. This sets up the environment variables for the build process.

Go to `PUBLIC\COMMON\OAK\CSP\X86\BIOSLOADER\LOADER`, open the `fat.h` file, and change the `READ_BUFFER_LENGTH_MAX` from `0x4000` to `0x3000`. This change is required to ensure BIOS memory for IDE mode detection will not be overwritten by the read buffer.

Rebuild the dependent library `iniparse.lib`, go to `PUBLIC\COMMON\OAK\CSP\X86\BIOSLOADER\INIPARSER` and rebuild that library.

Build the loader directory Then enter `build -c` on the command line. This should build the BLDR file in the same directory with a size of 20 KB.

- 2 Make a copy of the `SETUPDISK` floppy created in Section 3 and replace the BLDR in that floppy with the BLDR previously built with `READ_BUFFER_LENGTH_MAX` as `0x3000`.
- 3 Open `boot.ini` on the `SETUPDISK` copy and change the following parameters:

Change the `DisplayWidth`, `DisplayHeight` and `DisplayDepth` to suit the target screen resolution and color depth.

Change the line `BinFile=eboot.bix` to `BinFile=nk.bin`. This specifies which OS image to boot. We specify it as `nk.bin` and keep a working Windows CE OS image `nk.bin` in the Cactus device.

```
Set BakBinFile=eboot.bix
```

- 5 Use the updated `SETUPDISK` floppy to boot the target system / CEPC.
- 6 Copy the `boot.ini` file from `SETUPDISK` to the Cactus device. This file contains important boot parameters and is a ASCII text file editable using any standard text editor.
- 7 From the `BOOTDISK` floppy, copy `eboot.bix`, and `splash.bmx` to the device.

08 Troubleshooting Boot Issues

Windows CE BIOS boot loader uses serial port to dump error codes during the boot up process. `Loadcepc.exe` will also display any error messages. To display boot loader error codes via the serial port, connect the serial port of the target system with the development workstation with a serial cable, and configure the terminal program on the development workstation to the following:

```
COM1 384.k baud.  
8 data bits, 1 stop bit, no parity, no flow control.
```

There is an option to display full boot loader error messages in `debug.h` located in `WINCE500\PUBLIC\COMMON\OAK\CSP\X86\BIOS\LOADER\LOADER`. If the compiler directive `#define FULLMESSAGES` is enabled, then the full text corresponding to the error code will be displayed via the serial connection to the terminal.

Usually the biggest issue for booting Windows CE with the BIOS boot loader is using the Cactus device directly without recreating the partition table according to the current BIOS translation. The shipping partition table contains BIOS translation information that may be different from the current BIOS translation used on the target system. While the difference may not interfere with normal read/write operations, it may interfere with starting the boot loader at the correct location.

Another issue for booting is the use of imaging software to transfer the boot loader and kernel image to Cactus product. If the boot loader image is created from a product with different logical disk geometry, then the translation information contained in the partition table of the image will not match the actual configuration on the Cactus product. This difference may interfere with the boot process and cause boot failures.

Disk imaging software can be used for preparing mass production kernel images when the logical disk geometries for the image and the target devices are identical. This is the case when the final boot loader and kernel image is generated on the target to the Cactus product and the disk image is generated from the Cactus product and transferred to another identical Cactus product for production.

If you would like any additional information regarding data contained in this white paper feel free to contact a Cactus representative:

Asia/Pac RIM info@cactus-tech.com

Americas americas@cactus-tech.com

EMEA info@cactus-tech.com