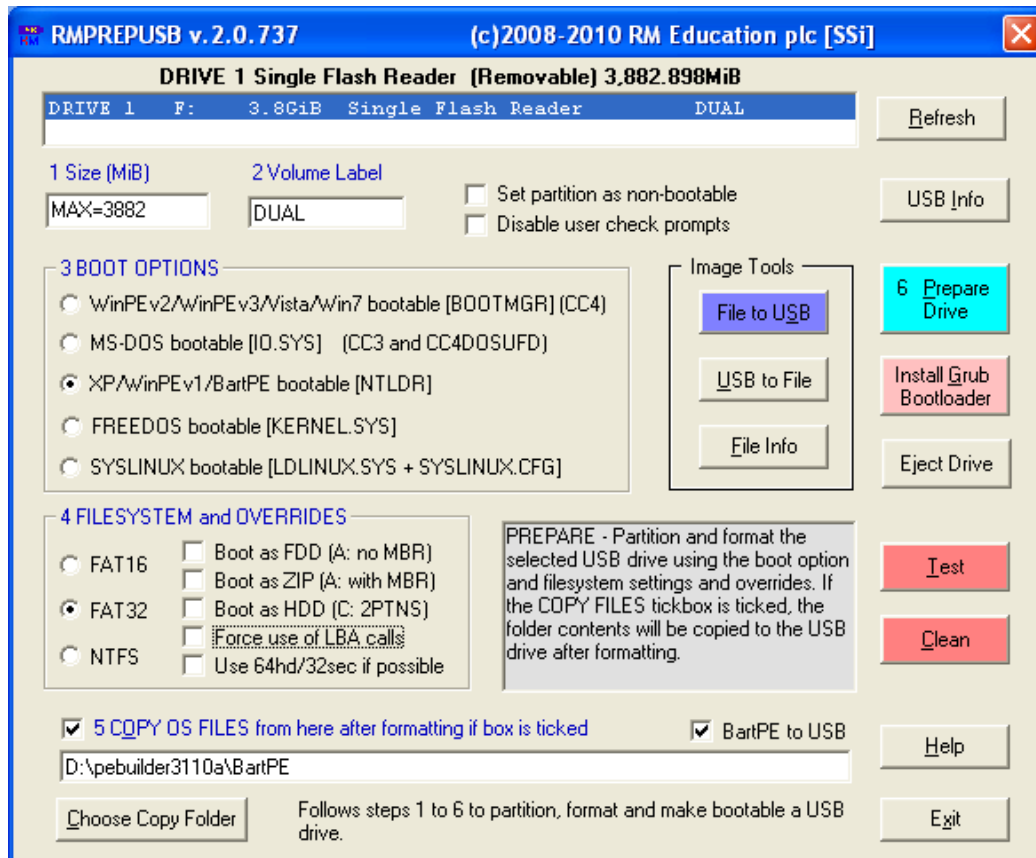


RMPrepUSB (v2.0.737 4 April 2010)

Introduction

RMPrepUSB is a Windows utility that can be used to format any USB storage device, e.g. USB Flash drive (UFD) or USB hard disk as a bootable device. It can be used to partition, format, write a Master Boot Record, partition table and/or Volume Boot Record (sometimes called a Partition Boot Record) and operating system boot code to any USB storage device (e.g. USB flash memory drive or USB hard disk).



Please read the FAQs at the end of this article!

Note: Coloured buttons will write to you USB drive. Except for the Install Grub button, all other coloured blue/red buttons will erase the USB drive contents. If you have any important files on the USB drive, copy them to a safe place first!

You do not need to install RMPrepUSB, just ensure that all files (RMPrepUSB.exe, RMPartUSB.exe, RMPrepUSB.pdf and MSVBVM60.DLL are the essential ones) are present in the same folder (it is recommended you do not use a folder on the Desktop or My Documents folder, use C:\RMPrepUSB for instance).

Important: always run these utilities with Administrator privileges

LICENCE SMALL PRINT:

RMPrepUSB and RMPartUSB are free (Freeware) **for private use only**; however they are not Open Source programs. For commercial use and licensing please contact support@rm.com. Distribution, sale or use in a commercial solution is forbidden without permission from RM Education plc (see FAQs below for more information).

RMPrepUSB can be used on USB Flash memory drive, USB hard disk drives or other USB storage devices.

RMPrepUSB has many different uses, including:

1. Partition, format and make bootable a USB drive (follow the blue steps 1-6)
2. Clean (erase/wipe) a USB drive
3. Backup a USB drive (or parts of a USB drive) to an image file
4. Restore an image file (or part of an image file) to a USB drive
5. Install the Grub bootloader onto a USB drive
6. Test a USB drive (useful for testing fake USB flash memory drives)
7. Eject a USB drive (for safe removal)
8. Display the partition structure of a USB drive (or image file)
9. Use after PEBuilder to install your XP PE files to a bootable USB drive.

When you click on one of the Prepare Drive, Test, Clean, Image Tool or Info buttons, RMPrepUSB will call the Windows command line utility **RMPartUSB** and the exact command line will be shown to the user. If you prefer, you can use RMPartUSB in your scripts or batch files rather than use RMPrepUSB. Type RMPartUSB from a Windows command line to see full instructions on how to use the utility or look at the RMPartUSB.txt readme file.

Note: The 'Install Grub' function and imaging operations ([FiletoUSB](#) and [USBtoFile](#)) are totally independent of any other settings within RMPrepUSB (i.e. the NTLDR, FAT32 and all other options have no affect when imaging – a binary image is simply copied to or read from the UFD, no additional 'tweaking' is done by RMPrepUSB or RMPartUSB). No extra files are copied afterwards.

Six steps to make your USB drive bootable

1. Select the USB drive in the listbox if more than one is present and either leave the size as MAX or change it (e.g. type 512 for a partition size of 512MiB).
2. Change the volume label text if you want to.
3. Set the type boot loader that you want installed onto the new partition after it has been formatted.
4. Change the Filesystem and Overrides. You may need to experiment with these before you can find a combination that works. Different models of PCs or Notebooks may require different settings.
5. If you want to, you can copy the contents of a folder to the new empty partition after it has been formatted. Tick the COPY OS FILES box if you want this. Note: RMPrepUSB will not place any boot files on the USB drive and it will not be bootable unless you copy the boot files onto it.
6. Click on the blue Prepare drive button.

Note: Before you unplug the USB drive, click on the Eject button to prevent file corruption.

Hint: Check your BIOS menu options! Some BIOSes have an option to boot a USB drive as a 'Fixed Disk' or a 'Removable Disk'. Try different BIOS settings if you are having difficulty getting your USB drive to boot!

About the COPY OS FILES option

RMPrepUSB prepares a USB drive by partitioning it, formatting it and placing boot code in the Master Boot Record (MBR) and Volume Boot Record (VBR, sometimes called the Partition Boot Record or PBR). The USB drive will not boot however unless you copy the required boot files onto the drive. You can automatically copy files to the USB drive after it has been formatted by ticking the COPY FILES option. If you do not tick this option then you must copy the boot files onto the USB drive manually afterwards.

How to boot to an Operating System

RMPrepUSB/RMPartUSB only partitions and formats the drive. You must provide boot files yourself. RMPrepUSB can copy these boot files to the USB drive if you use the COPY FILES folder and check box. Here are some quick tips on what you need.

Required extra boot files

- MS-DOS requires IO.SYS (or MSDOS.SYS depending on the version used) and COMMAND.COM in order to boot. You must obtain these files from somewhere (e.g. a DOS floppy boot diskette?). Simply copy these files onto the USB drive after you have formatted it using the MS-DOS option, by using the COPY FILES option to point to where you keep these MS-DOS boot files and tick the Copy checkbox.
- FreeDOS requires the files KERNEL.SYS and COMMAND.COM in order to boot. The latest version of RMPrepUSB includes the few essential boot files in a FreeDos folder, so simply use the COPY FILES option and select the FreeDOS option in RMPrepUSB.
- WinPE v1 and XP require many files to boot, but the first file is NTLDR. If you want to boot to BartPE, see Q24 below.
- WinPE v2 and v3 and Vista and Windows 7 require many files to boot, but the first file is BOOTMGR. For instance, just copy all files from a Vista or Windows 7 bootable DVD to the USB drive after setting the WinPEv2 option in RMPrepUSB.
- SysLinux will first look for the files LDLINUX.SYS and a SYSLINUX.CFG file. Many other files will also be required. If you are using isolinux, use the SysLinux RMPrepUSB option and then after the files have been copied over, rename isolinux.cfg to syslinux.cfg
- Grub will first look for the file GRLDR. RMPrepUSB will ask you if you want it to copy this file after it has installed the boot code.

If you want to use a different version of SysLinux or Grub, place your own versions of syslinux.exe, grubinst.exe and grldr into the same folder as RMPrepUSB and overwrite the ones provided.

Explanation of the Override options

The most complex part of RMPrepUSB is in selecting which overrides to use. The reason that there are so many choices is that different BIOSes behave in different ways. A USB Flash Drive (UFD) that is formatted using the HDD (2PTNS) option and containing MS-DOS boot file may boot as an A: drive on one system, but boot as a C: drive on another system. If you change the ***BIOS Setup menu options*** on the same system, it might then boot as an A: drive. On yet a third system, the same USB flash drive might not boot at all!

In addition, some BIOSes will not boot from a UFD at all if the physical drive size is greater than 512MB – or – some BIOSes may not boot from a UFD if the volume size is over (say) 1.3GB.

If you wish to boot from a UFD or USB hard drive, you may need to experiment with the settings below. You may find that one group of settings will work for one system and a different group of settings will work for another system. If using UFDs, always try a 512MB USB flash drive first before you move on to larger UFDs – some BIOSes will only work with smaller UFDs.

Boot as FDD (A: no MBR)

This option will format the drive with the chosen filesystem format option (FAT16\FAT32\NTFS) but the USB drive will have no partition table. The first sector of the USB drive will typically contain the operating system boot code. This option is usually used for an MS-DOS or FreeDOS drive that you wish to boot as a large floppy drive (i.e. they will boot to the A: prompt).

You can also select the 64hd/32sec option (recommended). If you de-select this option then the volume boot record will use 255 heads and 63 sectors per track.

Boot as ZIP (A: with MBR)

This option creates a Master Boot Record and partition table in the first sector of the drive. The Volume Boot Record code that is written to the drive will be suitable for booting MS-DOS or FreeDOS as a floppy drive. In addition, a drive geometry of 64hd\32sectors will be used (if possible).

If you wish to boot the USB drive as a hard drive then untick this option or tick the 64hd\32sec option.

‘Force use of LBA calls’ can be used with this override, but ticking the 64hd/32sec override will de-select the ZIP option, as the ZIP option will add the 64hd\32sec override.

Note: Some ZIP drive specifications define the first three entries in the partition table as unused and the fourth partition table entry is used, however this partition arrangement is not recognised by Windows and such a drive cannot be accessed by Windows Explorer.

Boot as HDD (C: 2PTNS)

This option simply adds a second, small hidden partition entry to the partition table. Some BIOSes will treat a USB drive as a hard disk if it sees more than one partition

table, because the ‘specification’ of a ZIP drive (super-floppy) is that it must only have one partition (thanks to ‘online’ of www.boot-land.net for this discovery). If you want to boot a system from a USB drive as a hard disk, set this option and untick the ZIP option.

You can try either the ‘Force LBA’ override or the 64hd\32sec override with this option.

Forcing the use of LBA calls

This option sets the Cylinder/Head/Sector values in the partition table to their maximum value of 1023 (3FEh) cylinders, 255 heads (FEh) and 63 sectors – even if the partition size is under 8GB. If BIOSes read a drive partition table and see that the CHS value is the maximum, then the BIOS may use a sector translation of 255 heads and 63 sectors per track. In addition, partition and volume boot record boot code will use LBA Extended Int 13h BIOS calls if it determines that the CHS values are set to the maximum. This also may help to successfully boot an operating system from a USB drive.

This option cannot be used with the 64hd\32sec option as they are mutually exclusive.

Use 64hd/32sec if possible

Instead of creating a partition table using the default drive geometry of 255 heads and 63 sectors per track, this option will use 64 heads and 32 sectors per track which is the most compatible setting for USB-ZIP (large-floppy) booting. If the partition is too large to use 64\32 geometry, then 128\32 will be used instead and then 255/32 and then 255/63.

This option is recommended for ZIP (large floppy) booting and FAT16.

Recommended settings

FreeDOS and MS-DOS boot as A:

Size=500 Boot.as.FDD 64hd\32sec FAT16

Size=500 Boot.as.ZIP 64hd\32sec FAT16

If either of these work, try using MAX size and/or FAT32

Ensure BIOS boot option for USB is set to ‘Removable Drive’ and not ‘Fixed Disk’

BartPE, WinPEv2\v3, FreeDOS and MSDOS boot as C:

Size=500 Boot.as.HDD. ForceLBA FAT16

Size=MAX Boot.as.HDD. (2PTNS) ForceLBA FAT32

Ensure BIOS boot option for USB is set to ‘Fixed Disk’ and not ‘Removable Drive’

Try also various combinations of these options. Note that one setting may work for one computer but not for a different computer!

Testing your BIOS

Some special MBR files are provided in the TESTMBR folder. These allow you to test how your BIOS ‘sees’ a USB drive when you change various options in

RMPrepUSB. After you have used RMPrepUSB to format your USB drive, use the FileToUSB button to **copy the first sector only** from one of the MBR_xPTN.BIN files to your USB drive. Please read the ReadMe.txt file in the TESTMBR folder for more details.

Tip for UFD booting as a hard drive:

If you have a Lexar or Netac USB Flash drive, obtain the utility BootIt.exe (try a Google search). This utility has a 'Flip Removable Bit' button – click on this and then unplug the UFD, then plug it in again – if it has worked, your UFD will be classified as a 'Fixed Disk' rather than 'Removable Media'. The utility only works with certain types of UFD controller chips. Many BIOSes will always boot a drive as C: (hard disk) if the device identifies itself as a 'Fixed Disk' rather than 'Removable Media'.

Bootling linux

To boot a linux system, use the SYSLINUX option in RMPrepUSB. This will format the UFD as a FAT16 or FAT32 MSDOS volume and then (after copying files if you tick the file copy option) will run syslinux.exe on the new UFD volume. The Windows version of syslinux.exe must be in the same folder as RMPrepUSB.exe. Syslinux.exe will install syslinux boot code into the volume specified and place the file ldlinux.sys on the UFD. Do not delete, overwrite or move this file once syslinux has placed it on the UFD or the UFD may not boot – it must remain at the same sector position on the UFD. If you already use syslinux.exe, replace the version in the RMPrepUSB folder with your version to avoid compatibility issues.

When the UFD boots, it will load ldlinux.sys and look for a syslinux.cfg file, which can contain a menu and details of which kernel to load. If you have an isolinux.cfg, you need to rename this to syslinux.cfg. More information can be found on the internet about syslinux.

If the UFD fails to boot after a while, either re-format the UFD using RMPrepUSB again or run syslinux.exe from the Windows command line (cmd) – e.g.

syslinux -fma F: (where F: is the volume letter of the UFD)

WARNING: take care not to specify your hard disk drive letter or your hard disk will not boot to Windows!

Installing the Grub bootloader

The 'Install Grub' button will run grubinst on the selected USB drive and then ask the user if they wish to also copy the grldr file. The USB drive should be already partitioned and formatted before you click this button. If the grub MBR does not seem to be written to the USB drive (because it does not seem to load grub but boots to DOS) then try the Eject button before you click on the 'Install Grub' button (but you will have to copy over the grldr file manually after re-inserting the USB drive).

Note: RMPrepUSB will not allow you to run grubinst onto physical drive 0 for safety reasons to prevent you from making your Windows hard disk unbootable.

Explanation of how BIOSes boot from a USB device

Most BIOSes require the user to enable USB booting and set the boot order so that a USB device will boot before the internal hard disk (although some BIOSes have an F10 or F12 key function which will allow you to choose a boot device from a menu). In addition you can often set the USB speed to USB 1.1 speeds (Hi speed) or USB 2.0 speeds (Full speed). If you are having difficulty booting from a USB drive, it is recommended to use a rear USB port, set the BIOS to USB 1.1 speeds and change the boot order so that the USB drive is first in the menu.

Before you enter the BIOS menu, switch off the system, insert the USB drive and then switch on the system – then press the F1 or F2 or Esc or DEL key (depending on your particular system) to enter the BIOS Setup menu.

All BIOSes behave differently, however disk storage devices are always accessed via Int 13h calls to the BIOS. There are two types of ways to access a disk device via the BIOS – ‘floppy’ or ‘hard disk’ read or writes. The BIOS must determine whether to allow access to the USB drive that it detects as either a ‘floppy’ (which is accessed with Int 13h DL=00h) or a ‘hard disk’ (which is accessed with Int 13h DL=80h). The BIOS has to determine which of these two access methods (DL=0 or DL=80h) to ‘map’ the USB drive to.

Once a BIOS determines how to map the USB device (either as a floppy device or a hard disk device), it will read one sector from the USB device (512 bytes) into memory at address 07C0:0000h and then jump to that code. The last two bytes of the sector must end in the two bytes ‘55h’ and ‘AAh’ or it is not considered to be valid boot code. When the CPU starts to execute the code that has just been copied into memory at 07C0:0000h, the BIOS has also pre-set the DL CPU register to the correct device number (either 00h or 80h). The boot code can use this value in DL to load more sectors into memory from the USB device.

There are three ‘types’ of USB drive and the BIOS must try to determine which of the three different ways to access the USB drive is to be used:

1. USB-FDD (floppy disk device)

The BIOS maps the device to ‘floppy’ Int 13h DL=0. A USB device may be detected as a USB-FDD device if it has a device name similar to ‘TEAC floppy drive’ and a capacity of 1.44MB, or possibly if it has no valid partition table. The device does not need to be formatted as 1.44MB, you can have a 1GB FAT16 UFD with no MBR.

2. USB-HDD (hard disk device)

A BIOS maps the device to ‘hard disk’ Int 13h DL=80h. A USB device may be detected as a USB-HDD device if it has a partition table and has more than one partition entry in that partition table. If only one entry exists the BIOS could treat the device as a USB-ZIP drive or a USB-HDD drive. Sometimes a BIOS menu setting can be changed to alter this behaviour.

3. USB-ZIP (pseudo-super-floppy)

A BIOS maps the device to ‘floppy’ Int 13h DL=0. A USB device may be detected as a USB-ZIP device if the drive has a partition table containing only one

entry. The BIOS may also require the device to be under a certain size (e.g. 512MB or less) or have values in the partition table of 64hds\32secs.

In addition, when a BIOS boots from a device as a USB-ZIP device it does NOT load the first sector MBR code (LBA 0) into memory – instead the BIOS loads the Volume Boot Record code directly into memory and jumps to it by reading the single entry in the partition table to find the start position of the partition (a USB-ZIP drive can thus only contain one partition entry). The BIOS then maps the USB drive to the Int 13h DL=0 call BUT it will always add the partition start address to any access. Thus a BIOS Int 13h DL=0 call to read cylinder 0, head 0 sector 1 (LBA 0) will actually read the first sector of the partition. The real sector 1 (LBA address 0) cannot be accessed at all. For instance, say the USB-ZIP drive has a partition table with a single entry. The partition table entry indicates that the partition starts at sector 32, then the BIOS would load sector 32 into memory and start to execute the code there. Any attempt to read CHS 0\0\1 (LBA 0) via a floppy Int 13h DL=0 call (or Extended Int 13h call) from that point onwards, will actually read sector 32. If the boot code attempts to read CHS 0\1\1 (LBA 32) it will actually read the sector located at CHS 0\2\1 (LBA 64) as the BIOS will always add the partition start address on to any access. In this way, the boot code located at the start of the partition is identical to that found on a floppy disk which has no MBR or partition table.

Some BIOSes do not support this type of booting and may always boot a USB-ZIP device as a hard disk (i.e. map the USB device to respond to Int 13h DL=80h calls). The difference between a USB-ZIP device and a USB-HDD device is very small, both have a master boot record and a valid partition table, however a USB-ZIP device should contain boot code that expects to be booting from a 'floppy drive' (DL=0) whereas a USB-HDD device should contain boot code that expects to be booting from a 'hard disk drive' (DL=80h). Many modern BIOSes contain a special menu option to change this behavior.

Frequently Asked Questions

Q1 – What happens if I don't tick any of the Override options (as in the screenshot above)?

A1 – A partition table will be created using a drive geometry of 255 heads and 63 sectors per track. Only one partition table entry will be made. The volume boot record will be created with hard disk compatible parameters.

Q2 – What code is used for the Master Boot Record (MBR) ?

A2 – The code used is identical to that used by Vista. It includes BitLocker support so that if you install Vista or Windows 7 onto a USB hard disk and then enable BitLocker, it should (!!!) work. The MBR will boot both 'floppy' USB drives and 'hard disk' USB drives.

Q3 – What boot code is used for the Volume Boot Record?

A3 – The VBR code will vary depending on what filesystem and format options you have selected, as follows:

FAT16 + MSDOS – MSDOS5.0 identifier, looks for IO.SYS, MSDOS.SYS (or WINBOOT.SYS). Possible Error Messages: **Error!**

FAT16 + MSDOS + ZIP – MSWIN4.1 identifier, with floppy disk parameters, looks for IO.SYS, MSDOS.SYS (or WINBOOT.SYS). Possible Error Messages: **Invalid system disk, Disk I/O error, Replace the disk, and then press any key.**

FAT16 + FreeDOS – FRDOS4.1 which boots KERNEL.SYS (if ZIP option used then floppy disk parameters are used) – Possible Error Messages: **Error**

FAT16 + BOOTMGR – uses MSDOS5.0 identifier and boot code and hard disk parameters. Possible Error Messages: **Remove disks or other media, Disk error, Press any key to restart.**

FAT16 + NTLDR – uses MSDOS5.0 identifier and boot code with hard disk parameters. Possible Error Messages: **NTLDR is missing, Disk error, Press any key to restart.**

FAT32 + MSDOS – MSWIN4.1 identifier and boot code which looks for IO.SYS and MSDOS.SYS (or WINBOOT.SYS). If ZIP option is used then uses floppy disk parameters. Possible Error Messages: **Invalid system disk, Disk I/O error, Replace the disk and then press any key**

FAT32 + FreeDOS – FREEDOS which boots KERNEL.SYS. If ZIP option is used then uses floppy disk parameters. Possible Error Messages: **Loading FreeDOS, No KERNEL.SYS**

FAT32 + NTLDR – uses MSDOS5.0 identifier and boot code and hard disk parameters. Includes 4 byte PATCH at E6-E9 changed to 90h (NOP). Possible Error Messages: **Remove disks or other media, Disk error, Press any key to restart.**

FAT32 + BOOTMGR – MSDOS5.0 identifier. Possible Error Messages: **BOOTMGR is missing, Disk error, Press any key to restart**

NTFS + BOOTMGR – Vista NTFS boot code. Possible Error Messages: **A disk read error occurred, BOOTMGR is missing, BOOTMGR is compressed, Press Ctrl+Alt+Del to restart**

NTFS + NTLDR – XP boot code with 4 byte PATCH at D9-DCh changed to 90h (NOP). Possible Error Messages: **A disk read error occurred, NTLDR is missing, NTLDR is compressed, Press Ctrl+Alt+Del to restart.**

Q4 – How do I copy a Vista or Win7 OS install DVD onto a UFD and make it bootable?

A4 – Select the BOOTMGR option + HDD (2PTNs). Browse to the OS DVD and set the DVD drive letter as the file copy path. Tick the ‘Copy’ checkbox. Click Prepare Drive.

Q5 – How do I copy an XP OS install CD onto a UFD and make it bootable?

A5 – There is no simple way to do this and make it bootable. Please visit www.boot-land.net for more information. If you wish to install XP onto the hard disk of a PC by booting from a UFD, you can use WinSetupFromUSB (which has a version of RMPrepUSB built in). If you want to boot to BartPE use PE-Builder to create a BartPE folder and then select this folder in RMPrepUSB and tick the COPY FILES box and the ‘PE to USB’ box.

Q6 – Why does my bootable BartPE\WinPE v2 (Vista) UFD boot on one system but not on another?

A6 – These operating systems expect be present on a ‘hard disk’ type of device which is accessed via Int 13h DL=80h. The BIOS that does not boot successfully is probably treating the UFD as a USB-FDD or USB-ZIP device and this ‘mapping’ the UFD to Int13h DL=00h and treating the device as a floppy device. It may be possible to change a setting in the BIOS setup menu (e.g. Emulate USB device as ‘Fixed Disk’) so that the UFD is treated as a hard disk. You should also try setting the HDD (2PTNs) option and the ForceLBA option. Also try changing the filesystem (e.g. use NTFS instead of FAT16). Some BIOSes will treat UFDs below a certain physical size (not partition size) as a USB-ZIP device, so try a 4GB or larger UFD. Also, try copying the boot files to the drive before the rest of the files. The best way to do this is to create a separate folder containing just the boot files (e.g. ntldr, ntdetect.com, boot.ini for XP, or IO.SYS, MSDOS.SYS and COMMAND.COM for MS-DOS or KERNEL.SYS for FreeDOS).

Q7 – Why does my MS-DOS bootable UFD boot as drive C: on some systems?

A7 – This will be due to the BIOS mapping the UFD as a hard drive (accessible via Int 13h DL=80h) rather than as a floppy device. Try the Boot as FDD and Boot as ZIP override options.

Q7 – I can boot from a FAT16 2GB UFD formatted as FAT16 but not from a 3GB or bigger UFD – why not?

A7 – MS-DOS cannot access FAT16 partitions over 2GB (Size=2000MB in RMPrepUSB/RMPartUSB). Set the partition size to 2000 or use FAT32.

Q8 – My PC seems to treat UFDs as a floppy device, how can I boot to WinPE v2?

A8 – There are ways to work around this. Please visit www.boot-land.net for solutions. A typical way is to use Grub or Grub4DOS as a 2nd stage boot loader which in turn will load the operating system. For instance, you can boot to a WinPE v2 iso file using Grub4DOS if your BIOS does not support USB-HDD booting.

Q9 – Where can I ask a question about RMPrepUSB or ask for help?

A9 – www.boot-land.net – post specific questions about RMPrepUSB in the Project Forge – USB formatting utilities – RMPrepUSB topic, for other questions choose the most appropriate forum and start a new topic.

Q10 – Can I re-distribute RMPrepUSB or use it as part of a commercial solution?

A10 – You can, but only with the permission of the author or RM Education plc (UK). This is because although RMPrepUSB and RMPartUSB are Freeware and made freely available for private use, it must not be used for commercial purposes or sold without permission. Any re-distribution must make this clear and contain some sort of ReadMe file or other notice containing this information as well as this pdf file. Commercial enquiries should be directed to support@rm.com, all other enquiries to the RMPrepUSB topic at www.boot-land.net. RMPrepUSB and RMPartUSB are the intellectual property of RM Education plc (www.rm.com). RM may ask you to remove the software or take legal action against you if these utilities are used for commercial purposes (i.e. if you are selling it) without permission having first been obtained (and after all, it's only polite ☺).

Q11 – Does RMPrepUSB/RMPartUSB work under WinPE v2/v3?

A11 – Mostly! You will need to add the WMI and scripting and XML components. Most functions will work except perhaps the folder browse button in RMPrepUSB.

Q12 – How can I use RMPrepUSB in my script/batch files?

A12 – You cannot, but you can use RMPartUSB. Type 'RMPartUSB' in a Windows command console window for help. RMPartUSB uses a command line and returns an errorlevel after completion. You can also use the LIST and FIND commands to find the drive number of a USB device. Tip: RMPrepUSB shows the exact command line that it uses to run RMPartUSB. For instance, you can use RMPartUSB in a cmd script to detect the presence of UFDs and partition and format them.

Q13 – Sometimes after running RMPartUSB, it returns a large negative error number – why?

A13 – This has been observed to occur when using Windows 2000 or when terminating the RMPartUSB command console window before RMPartUSB has itself terminated. For instance, if you click the USBInfo button in RMPrepUSB and then terminate the shell windows that is displayed by RMPartUSB, RMPrepUSB may report this error message. The reason is unknown.

Q14 – I used RMPrepUSB to format my UFD and make it bootable but there are no files on it and it won't boot?

A14 – RMPrepUSB does not contain or copy any boot files unless you provide them. Microsoft boot files are part of the operating system and cannot be distributed. You can obtain them if you are licensed to do so, or use Freeware like FreeDOS.

Q15 – I have set the USB partition size text box to 500 MB, how do I set it back to the maximum size?

A15 – Type 'MAX' in the Size text box or click on the Refresh button.

Q16 – RMPrepUSB reports an error when it runs – why?

A16 – Under Vista or Windows 7 you must right-click on RMPrepUSB.exe and choose ‘Run as Administrator’

Q17 – The Prepare button does not seem to work under Vista – nothing happens?

A17 – Try creating a folder at C:\RMPrepUSB – do not use a folder in special locations like the Desktop or My Documents as these may have special limited access rights under Vista/Win7.

Q18 – I used RMPrepUSB/RMPartUSB and now something is wrong with my USB pen and I can’t seem to access it.

A18 – Try the CLEAN and QUICKTEST option. If it tests OK, try to re-format it to a smaller size (e.g. 512MB). If that looks OK try a larger or maximum size. Use FAT16 or FAT32 (but not NTFS). Partition the USB pen to maximum size and then use H2TESTW to check the USB drive has no errors. If all else fails, try a different USB format utility.

Q19 – What does the QUICKTEST function do?

A19 – Quicktest will check a UFD for duplicate or bad ‘pages’. It writes to some sectors on the UFD and then reads them back to see if the information is correct. For example, it would write 0 to Sector 0 (first 8 bytes), then 2048 to the first 8 bytes at the 1Mb point (sector 2048), then 4096 to the first 8 bytes at sector 4096 (2Mb point), etc. – and then reads back the ‘marker’ values in each of the sectors. This is useful as a quick test of a UFD. If it passes this test then you should use H2TESTW to run a thorough test of all bytes.

Q20 – I ran QUICKTEST and it failed – what can I do to fix my UFD?

A20 – First you should realize that even if you manage to ‘fix’ the UFD **it will be very unreliable** and may fail (or corrupt the file contents on it) at any time. For instance, you could copy the latest draft of your new blockbuster novel to it, then copy it over the top of your master file on your hard disk. Then later you may find out that you have a corrupted file on your USB drive and have overwritten the original with the bad version – you have been warned!

One way to ‘fix’ it is to partition the UFD to a lower volume size. RMPrepUSB or H2TESTW will usually give you a clue as to what the maximum usable size might be. When you have decided on a size, proceed as follows (e.g. you have a 4GB pen but think only 1GB is usable).

1. Use RMPrepUSB to partition and format the UFD – use FAT32 or FAT16 rather than NTFS as accessing an NTFS filesystem that has errors or bad blocks can ‘upset’ Windows and lead to hangs/crashes. Set the size of the partition as required. Start low – e.g. 800MiB.
2. Run H2TESTW to test the new volume.
3. If the test passes, increase the size in step 1 – e.g. 900MiB and repeat steps 1 and 2 until H2TESTW fails.

Q21 – I started to use QUICKTEST and H2TESTW and at first I just got a few errors reported from my 4GiB UFD, but now I get write errors and the system seems to hang for ages during testing.

A21 – This is a typical symptom of cheap or ‘fake’ UFDs that have been constructed from faulty memory. The memory may be multiply mapped (e.g. 1GB mapped 4 times so it appears as 4GB) and the controller bad block management probably has not been set up properly. Error correction may not even be working correctly and the memory has probably not even been tested thoroughly by the UFD manufacturer. Everything may appear to be OK as long as you don’t use more than 1GB of the UFD, after that files will begin to get corrupted. In addition, more memory ‘pages’ may fail due to the stress of testing (or normal use) over time. If a UFD sector write fails, the PC can hang/freeze for many minutes on each sector write operation – be patient or pull out the UFD to make it fail quickly.

A UFD from a reliable make and source (Netac, Kingston, Lexar, etc.) can be tested many times without any problem.

Q22 – I have tested my UFD with RMPrepUSB and H2TESTW and it is fine, but after a week or so I get corrupt files or Windows wants me to reformat it and thinks it is faulty. If a re-format it then all is well until a few days later – is the UFD faulty?

A22 – Are you using the Windows SystemTray ‘Safely Remove Hardware’ icon and ‘ejecting’ the UFD before pulling it out of the USB port? When Windows sees a FAT filesystem, it loads the FAT tables into system memory. Any changes made to the files on the UFD will also cause the FAT tables in Windows memory to be updated – BUT NOT ALWAYS THE FAT TABLES ON THE UFD ITSELF! So if you simply pull out the UFD, the FAT tables on it will not have been updated. By using the ‘Safely Remove Hardware’ feature, you cause Windows to write all the new FAT table changes to the UFD. This is particularly important if your UFD appears as a ‘Fixed Disk’ to Windows and not as a ‘Removable Disk’. You can check to see if caching is used by right-clicking on the UFD drive in Windows Explorer and choosing Properties – Hardware – Properties – Policies. Even if it is set to ‘Quick Removal’ on your PC, you should always use the ‘Safely Remove Hardware’ icon on other PCs - so get into the habit and always do it!

If your UFD is formatted with NTFS, again always use the ‘Safely Remove Hardware’ icon before pulling out the UFD. The danger is that Windows can write or read to the UFD at any time, and pulling out the UFD at exactly the same time can lead to data corruption.

If you are still having problems, try the UFDs which have ‘Ultra Stable Technology’ (e.g. Netac) or similar. These UFDs employ a buffering scheme which prevents corruption due to power loss when a UFD is pulled out during write operations.

Q23 – I am having trouble getting some systems to boot. A USB flash memory pen works on one system but not on another. Some pens work on one system, but a different pen with an identical image does not work on the same system! What is going on?

A23 – This is a typical problem and it all comes down to the **BIOS**. Read the explanation above about how BIOSes boot USB devices. Here are some issues that have been observed with some BIOSes:

1. The BIOS will not attempt to boot any USB device over 512MB – the USB pen LED will not even flash during system POST as no read request is ever made!
2. The BIOS will not attempt to boot any USB device over 1GB.

3. The BIOS will treat any USB device under xxx MB as a USB-ZIP device, and any device over xxx MB as a USB-HDD
4. The BIOS will treat any USB device as a USB-ZIP and always boot it as a floppy device
5. The BIOS will treat any USB device as a USB-HDD and always boot it as a hard disk device
6. The BIOS will boot a USB-FDD formatted device that has no MBR or partition table, but will not boot a USB-ZIP or USB-HDD device
7. The BIOS will boot any device with only a single partition table entry in the MBR as a floppy device, but will boot a device that has two partition table entries as a USB-HDD device
8. The BIOS setup menu can be use to make the BIOS always treat a USB device as a 'Removable' device (USB-ZIP) or a 'Fixed Disk' device (USB-HDD) depending on user choice.
9. The BIOS looks at the Volume Boot record to determine type of device (e.g. if it sees MSWIN4.1 then it treats the device as a USB-ZIP device).
10. Partition parameters – if the partition has 32hds and 16 sectors per track then treat it as a USB-ZIP, if 63 sectors then treat it as a hard disk.
11. If the end cylinder is 1023 (max) then treat it as a hard disk.

A BIOS can be made to boot the same device in different ways – e.g. USB-FDD, USB-KEY, USB-ZIP, USB-HDD – try each in turn and try different ways of formatting with RMPrepUSB.

Be methodical in your testing! There are 5 variables to consider FOR EACH TYPE OF PC as below:

1. **BIOS** settings
2. **Physical size** of USB device
3. **MBR** on USB device (no MBR as in a USB-FDD device or with an MBR and partition table as in USB-ZIP and USB-HDD devices)
4. **Filesystem** the volume is formatted with (NTFS, FAT32 or FAT16)
5. **Code in VBR** (i.e. if the code expects to access a floppy drive or a hard disk)

The last three (3-5) are determined by settings in RMPrepUSB, the first two must be changed by the user. I suggest using a 512MB USB pen and a 2GB USB pen for all testing and make sure they have LEDs so you can see if the BIOS is reading them. Also use FreeDos for all initial testing because FreeDos can boot to either a 'floppy' device or a 'hard disk' device without error (unlike some versions of MSDOS). Try different BIOS and RMPrepUSB options and write down all results (all error messages – see Q3 above, and boots as A: or C:).

PC Model and BIOS version (use latest version if possible!)

USB pen make, model, size

RMPrepUSB options used

BIOS settings used

Result

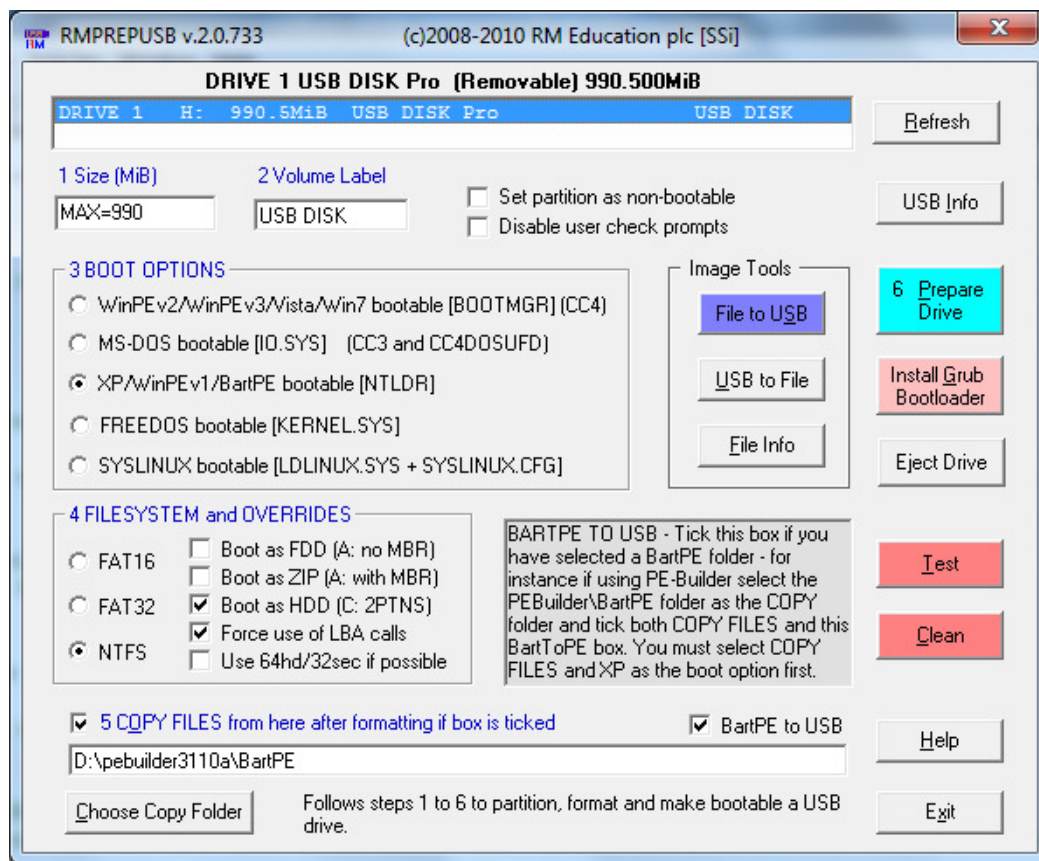
In between each test, switch off the system under test, then switch it on, then go into the BIOS and check the settings and that the USB pen is listed correctly. If you do not cold boot or reset, the BIOS may 'remember' the type of USB drive that was fitted previously even though you have re-partitioned it in between!

Remember, just because one type of system works with one particular USB pen drive, it does not mean that other systems will! Have fun and don't pull all your hair out!

Q24 - How can I make a bootable USB pen drive with BartPE on it?

A24 - Simple!

1. Obtain an XP installation CD
2. Download PE-Builder (e.g. v3110a)
3. Copy the contents of the XP CD to say C:\XPCD
4. Run PE-Builder.exe and select the source folder C:\XPCD
5. Check the Output folder is set to BartPE and Media Output set to None
6. Click Build
7. When finished run RMPrepUSB
8. Set the options as shown below (use FAT16, FAT32 or NTFS).
9. Click Prepare
10. When finished, click Eject
11. Job done!



Q25 - I want to copy my own extra files to my BartPE USB drive. Can I do this automatically?

A25 - Yes. When you tick the 'PE to USB' box, after the File Copy, RMPrepUSB calls the batch file RMPrepUSBXP.cmd. This batch file tweaks the files on the USB drive to make them bootable and also copies all files in the PetoUSB folder over to

the USB drive. So simply add your files to the PEtoUSB folder which is located in the same folder as RMPrepUSB. Alternatively, edit the batch file RMPrepUSBXP.cmd which is in the same folder as RMPrepUSB, to copy your own folder over. E.g.

```
xcopy /herky C:\myPEextras\*. * %pen%\*. *
```

Q26. I want to boot XP/WinPE v1 (not BartPE) – the ‘BartPE to USB’ function doesn’t seem to work with my version?

A26. You must look on the internet (Boot-Land) to find out how to prepare the source files correctly and how to convert the files so they will boot from a USB drive. Once you know how to alter the boot files, you can automate this by using Notepad to edit the file RMPrepUSBXP.cmd. This file is automatically run when you tick the **BartPE to USB** checkbox. If it is just a case of adding extra files to the USB key, you can simply drop these extra files into the PEtoUSB folder which is located in the same folder as RMPrepUSB. During the running of RMPrepUSBXP.cmd, any files and folders that are in the PEtoUSB folder will be copied across (and overwrite) any files on the USB drive.

Q27. I want to install BartPE on a USB drive but using Grub4DOS rather than NTLDR, but I cannot tick the ‘PE to USB’ checkbox when I select the MS-DOS option.

A27. This has been done to prevent misuse, however there is a deliberate loophole. First select XP/WinPE and tick the **COPY FILES** and **PE to USB** checkboxes. Then change the radio button option from XP/WinPE to either MS-DOS or FreeDOS. After RMPrepUSB has run, you can add your DOS boot files and Grub menu files.

--- o O o ---