

User manual

Boot manager Boot–US

Version 3.2.1

December/26/2013

Dr. Ulrich Straub

Copyright © 1999–2013

Table of Contents

1 Introduction....1
1.1 Overview....1
1.1.1 The package Boot–US....1
1.1.2 Central idea of Boot–US....2
1.1.3 Working with Boot–US....2
1.1.4 What Boot–US cannot do....3
1.2 Installation Boot–US (GUI)....4
1.2.1 Setup program for Boot–US (GUI)....4
1.2.2 Files in the installation directory....5
1.3 Update to new version....6
1.3.1 Update of configuration program Boot–US (GUI)....6
1.3.2 Update of configuration program Boot–US (command–line version)....6
1.3.3 Update of boot manager....6
1.4 Uninstallation....7
1.4.1 Uninstall configuration program Boot–US (GUI)....7
1.4.2 Uninstall configuration program Boot–US (command–line version)....7
1.4.3 Uninstall boot manager....7
1.5 Basic functions of Boot–US....9
1.5.1 Two Windows systems on the first disk....9
1.5.2 Two Windows systems on two disks....9
1.5.3 Windows and Linux on the first disk....9
1.5.4 Password protecting the entries in boot manager....10
1.5.5 Administrator password for Boot–US....10
1.5.6 Installation of boot manager on diskette or CD....10
1.6 Advanced function of Boot–US....11
1.6.1 Installing the boot manager on USB stick....11
1.6.2 Support for SSDs....11
1.6.3 Invisible start of boot manager....11
1.6.4 Installation of boot manager on any disk....11
1.6.5 One–time change of preselected entry in boot manager....11
1.7 Versions....12
1.8 Acknowledgement....21

2 Purchase a license....23
2.1 License....23
2.1.1 License agreement....23
2.1.2 Licensed functions....23
2.1.3 Testing all functions....23
2.1.4 DISCLAIMER OF WARRANTY....24
2.1.5 Distribution....25
2.2 Registration....25
2.2.1 License types and prices....25
2.2.2 Obtaining licenses online....25
2.2.3 License file....26
2.3 Installing the license....27
2.4 Contact....28



Table of Contents

3 Menu commands (GUI) 29
3.1 All menu commands of Boot–US 29
3.2 File / Login 31
3.3 File / Exit 31
3.4 Boot manager / Install 32
3.4.1 Boot manager / Install / Partitions (page 1 of 6) 32
3.4.2 Boot manager / Install / Password protection (page 2 of 6) 33
3.4.3 Boot manager / Install / Startup options (page 3a of 6) 33
3.4.4 Boot manager / Install / Startup options (page 3b of 6) 34
3.4.5 Boot manager / Install / Hide partitions (page 4 of 6) 34
3.4.6 Boot manager / Install / Installation target (page 5 of 6) 35
3.4.7 Boot manager / Install / Installation (page 6 of 6) 37
3.5 Boot manager / Remove 38
3.6 Boot manager / Show info 39
3.7 Partitions / Details 40
3.8 Partitions / Create 41
3.8.1 Partitions / Create / Select free space (Page 1 of 5) 41
3.8.2 Partitions / Create / Size of partition (Page 2 of 5) 41
3.8.3 Partitions / Create / Partition type (Page 3 of 5) 41
3.8.4 Partitions / Create / File system (Page 4 of 5) 41
3.8.5 Partitions / Create / Create partition (Page 5 of 5) 42
3.9 Partitions / Delete 43
3.9.1 Partitions / Delete / Select partition (Page 1 of 3) 43
3.9.2 Partitions / Delete / Method for deleting (Page 2 of 3) 43
3.9.3 Partitions / Delete / Delete partition (Page 3 of 3) 43
3.10 Partitions / Save sectors 44
3.10.1 Partitions / Save sectors / Select action (page 1) 44
3.10.2 Partitions / Save sectors / Save partition and boot sectors (page 2 of 2) 44
3.10.3 Partitions / Save sectors / Save track 0 (page 2 of 2) 45
3.11 Partitions / Restore sectors 46
3.11.1 Partitions / Restore sectors / Select action (page 1) 46
3.11.2 Partitions / Restore sectors / Restore partition and boot sectors (page 2 of 2) 46
3.11.3 Partitions / Restore sectors / Restore track 0 (page 2 of 2) 46
3.12 Configuration / Basic settings 48
3.12.1 Configuration / Basic settings / Language 48
3.12.2 Configuration / Basic settings / Password 48
3.12.3 Configuration / Basic settings / Boot disk 49
3.13 Configuration / Extended settings 51
3.13.1 Configuration / Extended settings / Trace 51
3.13.2 Configuration / Extended settings / Startup tip 51
3.13.3 Configuration / Extended settings / Backup 51
3.13.4 Configuration / Extended settings / Warnings 52
3.13.5 Configuration / Extended settings / Units 52
3.13.6 Configuration / Extended settings / Alignment 52
3.14 Window / Refresh 54
3.15 Help / Contents 55
3.16 Help / License 55
3.17 Help / Registration 55



Table of Contents

3 Menu commands (GUI)
3.18 Help / About Boot–US 55

4 Command–line version 57
4.1 Overview 57
4.2 Installation 58
4.3 INI file bootusc.ini for command–line program 59
4.3.1 Trace 59
4.3.2 Backup 59
4.3.3 Password for configuration program 60
4.3.4 Warnings 60
4.3.5 Use IEC units 61
4.3.6 Individual license texts in boot manager 61
4.4 Online help 62
4.5 Operations on sector level 63
4.5.1 Save track 0 63
4.5.2 Save sectors 63
4.5.3 Save partition and boot sectors 64
4.5.4 Restore track 0 64
4.5.5 Restore sectors 64
4.5.6 Restore partition and boot sectors 65
4.6 Operations on partition level 66
4.6.1 Hide partition 66
4.6.2 Unhide partitions 66
4.6.3 Activate partition 67
4.6.4 Deactivate partitions 67
4.7 Operations for boot manager 68
4.7.1 Install boot manager 68
4.7.2 Uninstall boot manager 71
4.7.3 Show status about installed boot manager 71
4.7.4 Update entry in boot manager 71
4.7.5 Preselect other entry in boot manager 73
4.8 Repair MBR 74
4.9 Report about partitions 75
4.10 Encrypt passwords 76
4.11 Reboot the computer 77
4.12 Exit codes of command–line program 78

5 Tips 79
5.1 In which cases one might face problems ? 80
5.2 Recommendations for safe usage of Boot–US 82
5.3 Uninstalling the boot manager Boot–US 84
5.4 Basic questions about Boot–US 86
5.5 Unhiding partitions in the boot manager 87
5.6 Error messages on integrity checks 88
5.7 Double occupied sectors in track 0 90
5.8 Limitations of different operating systems 91
5.9 Check independence and completeness 92



Table of Contents

5 Tips

5.10 Installing Windows multiple times on one disk....93
5.11 Installing Windows multiple times on different disks....94
5.12 Boot–US does not find SCSI or IDE disks under NT....95
5.13 Support for Windows 2000/XP/2003/Vista/2008/7/8....96
5.14 True partition hiding for Windows 2000/XP/2003/Vista/2008/7/8....97
5.15 Copying (cloning) a Windows 2000/XP/2003 installation....98
5.16 Administrator rights under Vista and Windows 7....99
5.17 Windows 7....100
5.18 Windows 8....101
5.19 Booting Linux by the boot manager of Boot–US....102
5.20 Linux partition is recognized by Boot–US as non–bootable....103

6 Glossary....105

6.1 Purpose of boot manager....106
6.2 Partition / partition sector / boot sector....107
6.3 Format of partition sector....108
6.4 Master Boot Record (MBR) / track 0....110
6.5 Primary / extended partition....111
6.6 Logical drive....112
6.7 Active partition....113
6.8 Hidden partition / true hidden partition....114
6.9 Bootable partition....115
6.10 Which partition IDs are recognized ?....116
6.11 LBA and CHS format, LBA mapping....118
6.12 INT 13h / extended INT 13h....119

7 Tech–Info....121

7.1 Structure of the configuration program Boot–US....122
7.2 Structure of the boot manager Boot–US....123
7.3 Standard boot process....124
7.4 Boot process with boot manager in primary partition....125
7.5 Boot process with boot manager in MBR....126
7.6 Boot process with boot manager on diskette....127
7.7 Booting from "second" disk....128
7.8 Uninstalling the boot manager Boot–US....129
7.9 Partition numbers....130



1 Introduction

1.1 Overview

1.1.1 The package Boot–US

The package Boot–US consists of a universal **boot manager** (fig. 1) and the accompanying **configuration program** for the boot manager.



Fig. 1: Boot manager Boot–US

The configuration program is supplied in two versions. The first is a Graphical User Interface (GUI) version for all WIN32/64 variants (95/98/ME/NT/2000/XP/2003/Vista/2008/7/8) (fig 2).

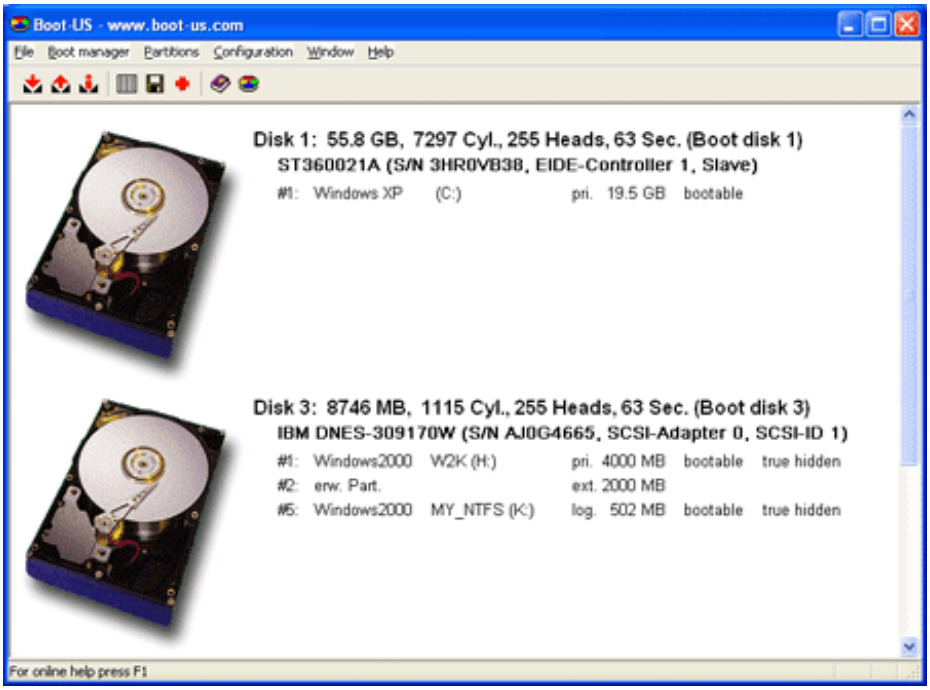


Fig. 2: Configuration program Boot–US (GUI)

In the second variant the configuration program is available also as command–line version (fig. 3) for DOS and all WIN32/64 variants.

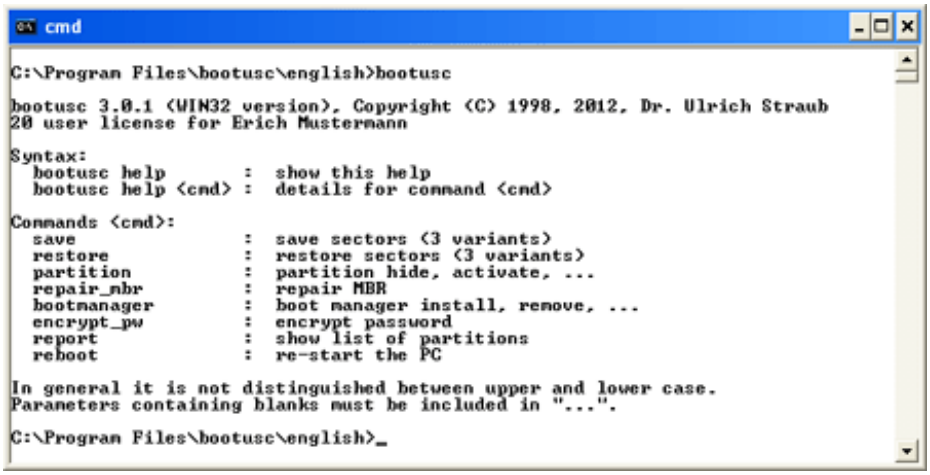


Fig. 3: Configuration program Boot–US (command–line version)

As a result of the separation in two programs the boot manager itself can be a small and simple program. The boot manager needs only handle the selection and booting of the desired operating system. The configuration program is the more complex component and, among other things, it is used to configure and install the boot manager. However, since the configuration program runs directly under Windows it offers the comfort of a graphical user interface. For example there are wizards guiding the user through the boot manager configuration process and there is extensive context sensitive online help.

The command–line version allows a script based installation of the boot manager. This is especially useful for companies with a large number of PCs. Additionally, there is a DOS version of the command–line program that can be installed on a DOS boot diskette. This allows configuration of the boot manager even if Windows is not running.

For a first contact with the package Boot–US it is recommended to use the GUI version. Therefore the command–line version is not further documented in this introductory chapter. The command–line version is fully documented in all details in an own chapter.

It is irrelevant which configuration program (GUI or command–line) is used to configure, install or remove the boot manager. The internal processing is the same, it is always the same boot manager which is installed. Furthermore all configuration programs can read older versions of the boot manager. When the configuration program is updated it is therefore not necessary to uninstall the boot manager before the update.

1.1.2 Central idea of Boot–US

Boot–US was developed in order to boot and run multiple operating systems completely **independently** of each other. By standard methods the desired operating systems must first be installed independently, see chapter tips. The later installation of the boot manager Boot–US can then be carried out with just a few mouse clicks. Once the boot manager of Boot–US is installed all the operating systems can comfortably be booted independent of each other.

1.1.3 Working with Boot–US

To use the Boot–US package the configuration program Boot–US (GUI or command–line) has to be installed first. For a first contact the Windows configuration program Boot–US is recommended. This program is a standard WIN32



program with a graphical user interface (GUI) and integrated online help. It can be installed on an any Windows partition (32 bit and 64 bit). For details please see the following chapter Installation Boot−US (GUI).

Only by the configuration program the boot manager of Boot−US is installed with the desired partitions (operating systems) to the hard disk (or CD or diskette), see menu command Boot manager / Install.... The installation wizard shows all bootable partitions on all local hard disk (EIDE or SCSI). A partition can be included in the boot manager with a simple mouse click. All necessary information −−− e.g. beginning, end, type of the partition and position of the boot sector −−− are determined and configured automatically by the configuration program. Thus the installation procedure of the boot manager does not require detailed system knowledge. Removing the boot manager is even simpler, see menu command Boot manager / Remove....

In addition to booting a partition from the hard disk, booting from diskette can be included in the boot manager of Boot−US. This allows you to set boot sequence to "C:, A:" and you can boot from diskette when required.

Boot−US allows you to save all disk sectors which are modified by the boot manager installation and operation, see menu command Partitions / Save sectors.... This includes all partition and boot sectors and the complete track 0 of all disks. It is strongly recommended to use this feature and to store the backup file additionally on a diskette. Of course the saved sectors can later be restored also, see menu command Partitions / Restore sectors....

Please read in any case the chapter tips for detailed instructions on a safe usage of Boot−US.

Boot−US shows all details of all partitions, see menu command Partitions / Details.... In this dialog it is also possible to hide/unhide partitions or activate/deactivate them. The menu command Partitions / Create... allows you to create partitions, while the menu command Partitions / Delete... allows to delete partitions.

The command−line version of Boot−US is intended mainly for companies with a large number of PCs. It allows a batch installation of the boot manager without any user intervention. Additionally the command−line version is also available for DOS, and thus it can be used even when no Windows is installed or when Windows cannot be accessed.

1.1.4 What Boot−US cannot do

- Boot−US cannot install or uninstall operating systems
- Boot−US cannot move or resize partitions
- Boot−US does not (fully) replace a partition manager
- Boot−US is not a disk editor
- Boot−US does not support booting by UEFI



1.2 Installation Boot−US (GUI)

Boot−US (GUI) −−− program name `bootus.exe` −−− is a standard WIN32 program. Its main purpose is to configure and install the boot manager. It is called the *«configuration program»* or the program *«Boot−US (GUI)»* or sometimes just *«Boot−US»*. This WIN32 program executes directly under Windows 95/98/ME/NT/2000/XP/2003/Vista/2008 (32 bit and 64 bit). It does not require switching to DOS mode.

1.2.1 Setup program for Boot−US (GUI)

The configuration program Boot−US (GUI) is installed by a standard Windows installation program. During the installation a few files (see below) are copied to the selected installation directory on the hard disk. The boot manager of Boot−US is **not** installed at this time. The boot manager of Boot−US must be installed explicitly from the configuration program Boot−US, see menu command Boot manager / Install.... During the installation of the configuration program Boot−US (GUI) entries in the Windows start menu are created pointing to the installation directory.

Under Windows NT/2000/XP/2003/Vista/2008/7/8 the configuration program Boot−US (GUI) does not require any additional files for execution. Under Windows 95/98/ME the two DLLs `disk32.dll` and `disk16.dll` are neccessary. These DLLs contain the functions for directly accessing the hard disk under Windows 95/98/ME. The DLLs reside in the same directory as the configuration program Boot−US (GUI). It is not necessary that these DLLs reside in the Windows system directory. Under Windows NT/2000/XP/2003/Vista/2008/7/8 the DLLs are also copied to the installation directory.

By default Boot−US uses only the installation directory for storing/creating files. All files and programs are searched or created in this directory. Also the INI file `bootus.ini` is searched and created in this directory. Boot−US stores all configuration settings in this INI file and not in the registry. The only exception is the password for the configuration program which can be stored in the registry if desired. During installation no DLLs in the Windows system directory are modified.



1.2.2 Files in the installation directory

In the installation directory you should find the following files:

File name	Comment
bootus.exe	configuration program Boot-US (GUI)
disk32.dll	direct disk access under Windows 95/98/ME (32-Bit part)
disk16.dll	direct disk access under Windows 95/98/ME (16-Bit part)
german\bootus_ger.chm	german online help for Boot-US (HTML Help format)
german\lizenz.txt	license information (german)
german\bestell.txt	information for ordering Boot-US (german)
german\zahlform.txt	pay form for ordering Boot-US (german)
english\bootus_eng.chm	english online help for Boot-US (HTML Help format)
english\bootus_eng.dll	english resources for Boot-US (GUI)
english\license.txt	license information (english)
english\order.txt	information for ordering Boot-US (english)
english\payform.txt	pay form for ordering Boot-US (english)
gksetup\cleanup.exe	additional clean-up when uninstalling Boot-US (GUI)
bootus.lic	license file (encrypted) for Boot-US
bootus.ini	INI file for Boot-US (GUI)

Remark: The last two files (license file and INI file) are optional.



1.3 Update to new version

1.3.1 Update of configuration program Boot−US (GUI)

A new version of the configuration program Boot−US (GUI) can simply be installed over an installed configuration program Boot−US (GUI). This replaces the configuration program and the online help. The current settings and a license file remain unchanged. It is **not** necessary to uninstall the configuration program first.

1.3.2 Update of configuration program Boot−US (command−line version)

A new version of the configuration program Boot−US (command−line version) can again simply be installed over an installed configuration program Boot−US (command−line version). This replaces the configuration program and the online help. The current settings and a license file remain unchanged. It is **not** necessary to uninstall the configuration program first.

1.3.3 Update of boot manager

In order to install a new version of the boot manager it is necessary to first install the new version of the configuration program (GUI or command−line version). After that you can install by the new configuration program the new version of the boot manager. It is **not** necessary to uninstall the boot manager first. The new boot manager is simply installed over an already existing boot manager.



1.4 Uninstallation

1.4.1 Uninstall configuration program Boot−US (GUI)

The simplest way to remove the configuration program Boot−US from the hard disk is by using the accompanying uninstaller. It removes the installation directory, the entries in the Windows Start menu and the entries in the Windows Install/Uninstall menu in the control panel. A potential INI file and a license file will not be deleted by the uninstallation command. Generally all files not created by the setup program remain unchanged by the uninstallation.

If necessary you can do the uninstallation also "manually". Since Boot−US has created only files in the installation directory it is sufficient to delete this directory in order to remove the whole package. Also all configuration settings −− with the potential exception of the password −− have been stored intentionally in the INI file and not in the registry. This all together makes a potential uninstallation very easy. When you manually uninstall Boot−US please do not forget to delete the entries in the Windows Start menu and the entry in the Install/Uninstall menu. The latter entry can be found in the registry under the key

`HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall`.

The password for the configuration program is stored optionally in the registry under the key

`HKEY_LOCAL_MACHINE\Software\ustraub\Boot-US`

In case the boot manager is installed on the hard disk, it is **not uninstalled** when the configuration program Boot−US (GUI) is removed. The boot manager must be removed explicitly by the WIN32 configuration program Boot−US, see menu command Boot manager / Remove....

1.4.2 Uninstall configuration program Boot−US (command−line version)

The configuration program Boot−US (command−line version) has been installed manually. Hence the uninstallation has also to be done manually, i.e. the corresponding files and folders must be deleted by the user. The installation process for the command−line version did not create any entries in the registry or the Windows Start menu, hence none of them must be deleted by the user in the uninstallation process.

In case the boot manager is installed on the hard disk, it is **not uninstalled** when the configuration program Boot−US (command−line) is removed. The boot manager must be removed explicitly by executing the command bootusc bootmanager remove.

1.4.3 Uninstall boot manager

There are three standard ways to uninstall the boot manager:

- Command "Boot manager / Remove..." of Boot−US (GUI)
- Command "bootusc bootmanager remove" of Boot−US (command−line version)
- Command "Uninstall boot manager" directly in the boot manager

In all three cases internally the same operations are carried out. More details on the uninstallation of the boot manager can be found in the chapter tips under Uninstalling the boot manager Boot−US.



Please note that uninstalling the boot manager does not affect the configuration program. The configuration program (GUI or command–line) remains unchanged when the boot manager is removed.



1.5 Basic functions of Boot−US

In this chapter we show a few simple examples for using the boot manager Boot−US. Since Boot−US is a universal boot manager there are many more combinations of operating systems which are bootable by the boot manager Boot−US. In this chapter we restrict ourselves on a few typical examples, and show where the boot manager Boot−US comes in. Additional and detailed installation instructions are found in the chapter tips.

1.5.1 Two Windows systems on the first disk

The aim is to run both Windows systems completely independent of each other. This can be achieved as follows:

- It is assumed that the first Windows system is already installed on a primary partition.
- The configuration program Boot−US (GUI) is installed on the first Windows system. A second primary partition for the second Windows system is created, formated and set active.
- The primary partition containing the first Windows is true hid by the Boot−US (GUI).
- The machine is rebooted and the installation of the second Window to the second partition is carried out. Since the first Windows is true hidden, the second installation proceeds like a new installation. Importantly, the true hidden partition remains unchanged by the second installation.
- The second Windows is booted and the configuration program Boot−US (GUI) is installed also on the second Windows system.
- The boot manager Boot−US is installed. Both Windows partitions are added to the boot manager. To hide partitions, the "automatically true hide primary partitions" option must be chosen.
- On the next start the boot manager Boot−US appears and offers both Windows systems.
- When you boot one of the two Windows systems the corresponding partition will be made visible automatically by the boot manager. The other Windows partition will be true hid automatically.

1.5.2 Two Windows systems on two disks

Again both Windows systems should be completele independent of each other. It does not matter which Windows system resides on which disk since Boot−US can boot all Windows versions from all disks.

- It is assumed that the first Windows system is installed already on a primary partition.
- The second disk must temporarily become the boot disk 1. This can be achieved by detaching the first disk or by changing the boot disk order in the BIOS.
- The second Windows is installed on a primary partition of the second disk (now boot disk 1). The second installation proceeds like a new installation since no Windows system exists on this disk.
- The previous disk order is restored.
- The configuration program Boot−US (GUI) is installed on the first Windows system.
- The boot manager Boot−US is installed. Both Windows partitions are added to the boot manager. To hide partitions, the "automatically true hide primary partitions" option must be chosen.
- On the next start the boot manager Boot−US appears and offers both Windows systems.
- When you boot one of the two Windows systems the corresponding partition will be made visible automatically by the boot manager. The other Windows partition will be true hid automatically.

1.5.3 Windows and Linux on the first disk

The following steps are necessary to setup this configuration:

- Windows is installed in a primary partition.



- Linux is installed in another primary partition. A linux loader must be installed into the boot sector of the linux partition.
- Both systems are now bootable by setting the corresponding partition active.
- The configuration program Boot−US (GUI) is installed on the Windows partition.
- The boot manager Boot−US is installed on the disk. Both systems are added. No hiding of partition is required.
- On the next start the boot manager Boot−US appears and offers both operating systems.

1.5.4 Password protecting the entries in boot manager

- When installing the boot manager you can specify individual passwords for the entries.
- Before booting an entry the respective password must be entered.

1.5.5 Administrator password for Boot−US

- You can set an administrator password when installing the boot manager.
- The administator password must be enetered on each start of the configuration program Boot−US.
- Inside the boot manager the administator password is required only for configuration changes.

1.5.6 Installation of boot manager on diskette or CD

- You can easily install the boot manager on diskette.
- When installing to CD an iso file is created. This iso file can then be used to create a CD.
- When booting from diskette or CD the configured boot manager is displayed.



1.6 Advanced function of Boot−US

In this section some advanced functions of the package Boot−US are presented.

1.6.1 Installing the boot manager on USB stick

- In the configuration program Boot−US (GUI) you can mark the USB stick as boot disk 1.
- Installation of the boot manager writes the boot manager always to the boot disk 1, which is the USB stick in this case.
- In the BIOS you need to change the boot disk order such that booting from USB comes first. If the USB stick is plugged in during the next startup the boot manager Boot−US will be displayed.

1.6.2 Support for SSDs

In order to take advantage of the full speed of SSDs partitions should be aligned on 1 MiB boundaries.

- Boot−US now supports partiton alignment on 1 MiB boundaries.
- The old convention of aligning partition on cylinder boundaries should not be used for SSDs.
- In the boot manager only sectors are written to disk if their contents has changed.

You can check the current alignment of partitions easily. Just turn on the warnings by the command `"configuration/ext. settings..."` and turn on and off the assumed aligment on 1 MiB.

1.6.3 Invisible start of boot manager

- The boot manager can be invisible when it is started. It is active in the background.
- In the invisible mode the preselected partition is booted.
- In order to select another entry the boot manager has to be made visible by pressing the pre−defined hotkey.
- There are many key combinations defineable as hotkey.

1.6.4 Installation of boot manager on any disk

- The boot manager can be installed on any disk.
- In the configuration program Boot−US (GUI) you only need to mark the respective disk as boot disk 1, and then install the boot manager.
- When starting the pc the specified disk must be the boot disk 1 otherwise the boot manager does not appear.

1.6.5 One−time change of preselected entry in boot manager

- For maintenance purposes it is often required to boot a certain partition just once.
- Afterwards the usual partition should be booted again.
- This one−time change of the preselected entry is available since version 3.2.0. The configuration is done by the command−line version of Boot−US.



1.7 Versions

Boot−US 3.2.1: created on December/26/2013
Improvements:
− license check removed for command−line functions 'bootmanager status' and 'bootmanager preselect'
Fixes:
− none

Boot−US 3.2.0: created on December/08/2013
Improvements:
− one−time boot of an entry in the boot manager
Fixes:
− none

Boot−US 3.1.0: created on January/15/2013
Improvements:
− support for Windows 8
− support for cursor keys on numpad
Fixes:
− display of volume name for duplicated partition fixed

Boot−US 3.0.2: created on February/01/2012 (fixed on Feb/12/2012)
Improvements:
− none
Fixes:
− display of 11 and more entries in boot manager fixed
− wrong colors in boot manager by command−line installation fixed
− wrong creation of image file with 8 or more entries fixed (Feb/12/2012)

Boot−US 3.0.1: created on January/09/2012 (fixed on Jan/21/2012)
Improvements:
− IEC units are now the default
− ENTER key on numpad supported again
− special characters are allowed in password entry
Fixes:
− diskette installation fixed
− booting from disk 2 fixed
− characters z and y within password were interchanged (Jan/21/2012)

Boot−US 3.0.0: created on July/10/2011 (Beta version)
What's new?
The most important extensions are the support for SSDs, the installation of the boot manager into a non−active primary partition and the invisible start of the boot manager (see list below).
Improvements:
− partitions may be aligned on 1 MiB boundary (support for SSDs)
− boot manager does not rewrite unmodified sectors (support for SSDs)
− installation of boot manager into non−active primary partition
− invisible start of boot manager



− installation to CD creates now ISO file
− lba access is now the default setting
− command−line version allows to change the preselected partition
− partitions size displayed in SI or IEC units
− installation to diskette may now contain twice as many sectors
− automatic hiding in boot manager restricted to configured disks
− license texts in boot manager can now be changed individually
− export of boot manager configuration in file bmgrus.ini
Fixes:
− none

Boot−US 2.1.8: created on February/20/2010 (fixed on Aug/22/2010)
Improvements:
− installation in MBR can skip a sector occupied by third−party program
Fixes:
− installation on diskette did not work under Windows Vista and 7
− failed disk access is retried in old and new variant (Aug/22/2010)

Boot−US 2.1.7: created on November/08/2009 (fixed on Nov/15/2009)
Improvements:
− boot sector is not checked any more for disk parameters
Fixes:
− setup did not run under Windows 7 (Nov/15/2009)

Boot−US 2.1.6: created on March/13/2005 (fixed on April/24/2005)
Improvements:
− none
Fixes:
− installation of boot manager in primary partition beyond 8 GB fixed: parts of the boot manager could get overwritten some time later
− update operation of command−line version fixed (April/24/2005)

Boot−US 2.1.5: created on May/20/2004
Improvements:
− DOS command−line version now runs in protected mode
Fixes:
− writing and reading of *.sec file fixed
− formating and overwriting partition under Windows 9x/ME fixed

Boot−US 2.1.4: created on March/14/2004
Improvements:
− start signal can now be specified by all configuration programs
Fixes:
− check of FAT16 boot sector was a bit too strict
− partition name and description was not displayed in password dialog of boot manager

Boot−US 2.1.3: created on February/17/2004
Improvements:
− none



Fixes:
− check of FAT32 boot sector was a bit too strict
− wrong exit code of command "bootusc save partsec"

Boot−US 2.1.2: created on February/08/2004
Improvements:
− when deleting a partition it is possible to erase all sectors of the partition
− when creating a partition it is possible to format it as FAT12/FAT16/FAT32
− on true hiding an encrypted partition remains unchanged
− the Boot−US (GUI) remembers the window size and position
Fixes:
− several problems on deleting or creating a partition have been fixed
− password entry in login dialog under XP displayed unusual characters
− installation problem of boot manager on diskette has been fixed

Boot−US 2.1.0: created on June/03/2003 (Beta version)
Improvements:
− boot manager can be installed in primary partition beyond the 8 GB limit
− creating and deleting partitions can now be performed in the configuration program
− boot manager does not show an error when trying to read from removable disks without media (ZIP drive)
− user manual is provided as pdf file
− image file containing boot manager Boot−US can be created
− modernized appearance of configuration program Boot−US (GUI)
− size of dialogs and wizards depends on screen size
− integrity check does not show a warning any more for CHS cylinder number 1022
Fixes:
− none

Boot−US 2.0.6: created on December/09/2002
Improvements:
− partition ID of boot manager entry can be changed now
− better recognition of available administrator privileges of user
Fixes:
− none

Boot−US 2.0.5: created on November/27/2002
Improvements:
− none
Fixes:
− partition 0Eh was treated as a FAT32 partition (correct is FAT16)
− available support for ext. INT 13h was sometimes not recognized in DOS command−line version

Boot−US 2.0.4: created on August/29/2002 (fixed on Sept/01/2002)
Improvements:
− none
Fixes:
− booting of Windows NT/2000/XP/2003 from disk 2 failed when boot manager was installed to hard disk and the boot manager option "select standard partition" was set
− installation of boot manager to diskette could have been incomplete when individual hiding of partitions was used



− checking duplicated entries fixed (Sept/01/2002)

Boot−US 2.0.3: created on August/21/2002
Improvements:
− when the boot manager is installed in a primary partition the current MBR is saved within the primary partition, therefore the track 0 remains unchanged in this case
− when the boot manager is installed in MBR the main part of the boot manager is stored near the end of the track 0
− added tooltips for buttons on page 1 of installation dialog
− informations about all hard disks can now be displayed directly in the boot manager
− command−line version extended by reboot function
Fixes:
− drive letters are displayed again
− removed the DEBUG output of the licensed command−line program

Boot−US 2.0.2: created on June/22/2002
Improvements:
− configuration program (GUI) allows to duplicate entries in boot manager
− online help is now built in modern HTML Help format
Fixes:
− only Windows NT/2000/XP/2003 are booted from disk 2, 3, ... by means of disk swapping, for all other cases (e.g. Linux) the former method is used again
− selecting the former booted partition did not work in boot manager 2.0.1
− hiding of logical partitions did not work in the command−line version

Boot−US 2.0.1: created on April/26/2002 (Beta version)
Improvements:
− none
Fixes:
− booting of DOS and Windows 95/98 from disk 2, 3, ... did not work with the new method, hence for these cases the previous method is used again

Boot−US 2.0.0: created on April/15/2002 (Beta version)
Improvements:
− boot manager supports booting of Windows NT/2000/XP/2003 from disk 2, 3, ...
− order of boot manager entries is unchanged in new installation
− boot manager displays precise error messages with disk and sector number
− command−line version supports uninstallation of boot manager
− command−line version supports status info about installed boot manager
− command−line version supports update of separate entries without complete re−installation
− command−line version supports installation of boot manager with simplified configuration menu (F10 key)
− configurable acoustic signal upon start of boot manager (F10 key)
Fixes:
− none

Boot−US 1.7.1: created on December/09/2001
Improvements:
− automatic hiding of pri. partitions now behaves identical to previous versions
− automatic hiding of pri. and log. partitions added as separate configuration possibility
Fixes:



− saving partition names in INI file failed if INI file was empty

Boot−US 1.7.0: created on October/14/2001 (Beta version)
Improvements:
− individual hiding of logical partitions
− warnings about invalid CHS values can be disabled
− command−line version of Boot−US has now same functionality as GUI version
− command−line version of Boot−US is available as DOS version and as WIN32 version
Fixes:
− installation of boot manager into primary partition did not re−activate already present Boot−US partition

---

Boot−US 1.6.3 (GUI): created on August/21/2001
Improvements:
− none
Fixes:
− storing parameters in INI file fixed
− IDE disk model is recognized also on Windows ME
− checking of CHS values in partition table was too strict
− invalid or incomplete formated NTFS partition is now handled correctly

Boot−US 1.6.2 (GUI): created on July/22/2001 (fixed on July/23/2001)
Improvements:
− improved speed of program execution with disabled trace
− volume label of hidden NTFS partitions is recognized
− more integrity checks (LBA mapping) of partitions
Fixes:
− handling of keys 'y' and 'z' when enterings passwords in boot manager fixed
− installation program GkSetup failed when a file 'setup.exe' was present in same directory
− message about violation of cylinder boundaries for logical partitions corrected (July/23/2001)

Boot−US 1.6.1 (GUI): created on June/17/2001
Improvements:
− integrity of partitions (overlapping partitions) is verified
− NTFS version is recognized
− volume label of hidden FAT16/FAT32 partitions is recognized
Fixes:
− empty warning could appear when installing non−licensed version 1.6.0

Boot−US 1.6.0 (GUI): created on May/30/2001 (Beta version)
Improvements:
− order of boot disks can be specified within configuration program
− disturbing disks can be skipped
− configuration program recognizes disk model and serial number
− booted partition is activated (if possible) by boot manager
− timer in boot manager is stopped on first key stroke
Fixes:
− none



Boot–US 1.5.4 (GUI): created on Jan/12/2001
Improvements:
– true hiding of partitions can now be tested when boot manager is installed to diskette
Fixes:
– none

Boot–US 1.5.3 (GUI): created on Dec/10/2000
Improvements:
– improved work around for Windows NT/2000/XP/2003 problems with diskette access
Fixes:
– none

Boot–US 1.5.2 (GUI): created on Dec/04/2000
Improvements:
– none
Fixes:
– individual true hiding did not work in the boot manager

Boot–US 1.5.1 (GUI): created on Nov/26/2000
Improvements:
– correct configuration of true hiding is checked when boot manager is installed
– work around for Windows NT/2000/XP/2003 problems with diskette access
– highest free partition table entry is used for boot manager partition
Fixes:
– none

Boot–US 1.5.0 (GUI): created on Oct/16/2000 (Beta version)
Improvements:
– true hiding of partitions (necessary especially for Windows 2000/XP/2003)
– all boot and partition sectors on all disks can be saved manually
– automatic saving of boot and partition sectors before write operations
– all partition names are stored additionally in INI file
– context menus in many dialogs
Fixes:
– administrator password for Boot–US could not be stored in boot manager due to version conflict

Boot–US 1.4.0 (GUI): created on June/29/2000
Improvements:
– new format of license file (registered license files remain valid)
Fixes:
– none

Boot–US 1.3.1 (GUI): created on June/02/2000 (fixed on June/15/2000)
Improvements:
– uninstallation now possible directly from boot manager
– contents of all sectors written is always verified now
– partition labels are obtained now directly from WIN32 functions
– before careless users ignoring warning could write empty MBR to disk
Fixes:



– problems with individual hiding of partitions fixed
– installation of boot manager to diskette fixed
– password entry for Windows program Boot−US fixed
– saving of last booted partition in boot manager fixed (June/15/2000)

Boot−US 1.3.0 (GUI): created on April/16/2000 (Beta version)
Improvements:
– individual hiding of primary and extended partitions
– configurable description for partitions in boot manager
– password protection for configuration program
– administrator password for boot manager
– drive access is repeated also under NT
– graphical design of boot manager reworked
– timer in boot manager can be deactivated completely
– boot manager remembers last booted partition
– empty disks are ignored by the boot manager
– boot manager avoids ext. INT 13h for better compatibility
– installation on diskette may occupy up to 36 sectors
Fixes:
– for disks with > 1023 cyl. and < 255 heads the part. size was sometimes displayed wrong

Boot−US 1.2.4 (GUI): created on Jan/01/2000
Improvements:
– support for Windows 2000/XP/2003
Fixes:
– none

Boot−US 1.2.3 (GUI): created on Oct/23/1999
Improvements:
– license file is Base64 encoded to prevent corruption by email transport
– total number of sectors returned by extended INT 13h is now used
Fixes:
– installing boot manager in primary partition sometimes required to uninstall it before

Boot−US 1.2.2 (GUI): created on July/11/1999
Improvements:
– complete english online help added (Sept/13/1999)
Fixes:
– boot manager could not be installed in MBR when partition table on disk 1 was full
– unchanged passwords were corrupted at second installation of boot manager

Boot−US 1.2.1 (GUI): created on July/03/1999
Improvements:
– explicit support for booting DOS, Windows 95/98/ME from 2nd, 3rd.. hard disk
– boot manager now passes appropriate disk number (80h, 81h...) in DL register
– password protection for partitions
– by default hiding of partition is not activated
– by default boot manager is installed in primary partition
– info about target partition is shown before boot manager is installed in primary partition



– drive access is retried in case of errors
– language in Boot−US and boot manager can be changed to English
– boot manager allows manual unhiding of all partitions
Fixes:
– determination of LBA mapping from BIOS corrected
– determination of partition parameters corrected
– determination of drive letter corrected

Boot−US 1.1.4 (GUI): created on April/13/1999
Improvements:
– determination of LBA mapping selectable (from BIOS or partition table)
– boot manager in primary partition occupies full cylinder for better compatibility
– new design for toolbar (flat look)
Fixes:
– no LBA access was used for part. 0Bh, 0Ch, 0Eh, 0Fh and disk size < 8 GB
– CHS or LBA values were wrong when LBA−CHS−mapping was not correct

Boot−US 1.1.3 (GUI): created on March/16/1999
Improvements:
– online help integrated into Boot−US
– Boot−US supports now old version of system DLL Comctl32.dll
Fixes:
– bug in VC++ list control caused problems with empty strings
– empty hard disks without partitions were not displayed correctly

Boot−US 1.1.0 (GUI): created on Jan/11/1999
Improvements:
– menu command to get info about installed boot manager
– partial editing of partition table possible
– automatic hiding of C: partitions can be disabled
Fixes:
– none

Boot−US 1.0.5 (GUI): created on Nov/29/1998
Improvements:
– boot manager can be installed in primary partition
– boot manager can automatically hide additional C: partitions
– display of license information
Fixes:
– none

Boot−US 1.0.4 (GUI): created on Nov/01/1998
Improvements:
– boot manager supports disks up to 2048 GB with extended INT 13h
– Windows 9x and NT are recognized from their boot sector
– trace output much faster than before
Fixes:
– none



Boot–US 1.0.3 (GUI): created on Oct/01/1998
Improvements:
– wizard for installation of boot manager
– complete track 0 can be saved and restored
– hidden partition (OS/2 boot manager) are recognized
– many additional partitions are recognized
– menu command for refreshing partition list
– support for trace output
Fixes:
– entries in log. partition sector were not recognized when in wrong order

Boot–US 1.0.0 (GUI): created on Sept/10/1998
– first version for general public

Boot–US 1.5.0 (cmd): created on Nov/19/2000
Improvements:
– true hidden partitions can be unhid
Fixes:
– none

Boot–US 1.3.1 (cmd): created on June/11/2000
Improvements:
– contents of all sectors written is always verified now
– multiple sector access replaced by single sector access
– disk access avoids ext. INT 13h whenever possible
Fixes:
– program did stop sometimes

Boot–US 1.3.0 (cmd): created on April/09/2000
Improvements:
– language can be changed to english or german
– hidden extended partitions can be unhid
Fixes:
– none

Boot–US 1.2.0 (cmd): created on June/28/1999
Improvements:
– Partitions can now be made visible on all disks
Fixes:
– Only first disk could be specified

Boot–US 1.1.5 (cmd): created on May/15/1999
– first version for general public



1.8 Acknowledgement

Users and testers:
First of all I like to thank all users who reported errors or suggested improvements or took part in the beta tests: *D. Arold, M. Bach, J. Boenisch, N. Brauer, H. Buchta, J. Cook, R. Curtis, A. Deinlein, D. Deutzer, J. Ebend, M. Ellinger, U.–P. Egger, K.–H. Fink, O. Foellmer, L. Forsblad, K. Friedrichs, Z. Gnilka, L. Gusewski, B. Hagstrand, F. Harbott, D. Herkner, D. Hettmann, A. Hoefler, F. Hoffmann, H. Huang, T. Huber, W. Huempfner, M. Jonas, T. Jacob, S. Kao, W. Kasdorp, M. Kirschner, C. Klug, B. Koch, N. Krismer, H. Lampe, T. Laun, S. Lindner, A. Maier, C. Marklund, K. R. Mathisen, S. McFee, O. Mishuris, S. Mueller, Dr. B. Niklaus, S. Nouri, G. Phieler, J. Pichler, H. F. Piotraschke, S. Raeuber, D. Rehm, K. Reich, J. Rensmann, J. Roehrenbeck, C. Rohrbach, M. Sauerhammer, B. Scheim, J. Schieck, R. Schillmann, T. Schlegel, T. Schroeder, S. Schulte, A. Schwingenheuer, R. Seggering, M. Seifert, F. Siebe, F. Steinkuhl, H. Tichy, Ing. H. Vogler, S. Waidele, S. Waldmann, A. Watzal, A. Weger, C. Wittmer, D. Zuber.*

Special thanks goes to Mike Wyatt for his numerous corrections of the english translation.

Windows program:
The user interface of the window program has been developed with the Visual C++ MFC library. Many interesting details of the user interface originate from "MFC Programmer's SourceBook" (www.codeguru.com). The willingness of all authors to allow other programmers to use their source code is gratefully acknowledged. The articles of the following people have been especially helpful:

- Brent Corkum (menu with bitmaps)
- Joerg Koenig (flat toolbars)
- Zafir Anjum (edit subitems in list control)

Installation program:
The setup program for Boot–US has been created by the freeware tool "GkSetup" of Gero Kuehn (www.gkware.com).





2 Purchase a license

2.1 License

2.1.1 License agreement

The whole package Boot−US (GUI and DOS command−line version) with all accompanying programs (boot manager, etc.) may be used in the non−licensed version on **private** PCs for free. This allows a broad usage of the program. Public schools may also use the non−licensed version of Boot−US for free.

The report function of the command−line version of Boot−US (for DOS and WIN32) may be used also in the non−licensed version for free. This function has been intentionally not protected by a license.

Additionally, the whole package Boot−US may be tested in the non−licensed version during a trial period of one month for free. If you want to use Boot−US after this trial period on **commercially** used PCs, you must purchase a license. The license purchased entitles the licensee to use Boot−US on the corresponding number of PCs and it entitles the licensee to use the forthcoming versions of Boot−US for free.

All future versions of Boot−US together with a (old) license allows to use the previously available functions. Please note that **new functions in future versions** of Boot−US might not be available with an old license. The author reserves the right to determine whether new functions in future versions are available free of charge, or are available with an old license or do require a new license.

2.1.2 Licensed functions

The following functions require a license:

Version	Funktion
Version 3.x.x	Complete configuration of invisible start
Version 2.x.x	Password protection in boot manager
Version 2.x.x	Full password protection for configuration program
Version 2.x.x	Support for boot from second, third,... disk
Version 2.x.x	Support for true partition hiding in boot manager

When the boot manager is installed to the hard disk these features are available only in the **licensed** version.

The WIN32 command−line version is intended especially for larger installations. The installation and uninstallation of the boot manager requires at least a 20−PC license. All other functions (except the report function) require at least a single PC license. Only the report function may be used without a license.

The 100−PC license (and higher) allows to individually change the license texts displayed in the boot manager.

2.1.3 Testing all functions

It is possible to try all functions of the package Boot−US before purchasing a license. To do this it is only necessary to install the boot manager to diskette or CD. When the boot manager is installed to **diskette** or **CD** all features are also



available in the non−licensed version. This allows the potential customer to check and **try out all** the features of Boot−US.

2.1.4 DISCLAIMER OF WARRANTY

The whole package Boot−US has been developed with great care and has undergone extensive testing. But it **cannot** be guaranteed that all functions of Boot−US are fully operational and work as described, and it cannot be guaranteed that Boot−US is free of errors. You are allowed to test Boot−US freely for one month in order to verify whether Boot−US works reliably and without errors in your hardware environment. At last, we would say that you are using Boot−US **at your own risk**.

Please also read the legal stuff below. By using Boot−US you accept these conditions of usage. If you do not agree you are not allowed to use Boot−US.

The installation program Boot−US allows you to perform a test installation of the boot manager to diskette and it allows to save the whole track 0 of the hard disk. Both measures support a step−by−step testing of Boot−US in the actual hardware environment. It is expected that during such a step−by−step installation of the boot manager potential problems become quickly evident and can also be repaired completely. Additionally the DOS program **bootusc.exe** has been developed for repairing a failed installation even at a plain DOS command prompt.

If you detect problems or if you have ideas for improvements, please inform the author. It is intended to correct problems as fast as possible (hopefully within a few days). On the other hand, improvements are expected to require more time and will therefore be implemented only in the forthcoming versions.

Legal stuff:
This software is provided on an "AS IS" basis, without warranty of any kind, including without limitation the warranties of merchantability, fitness for a particular purpose and non−infringement. The entire risk as to the quality and performance of the software is borne by you. Should the software prove defective, you alone assume the entire cost of any service and repair.

SOME STATES DO NOT ALLOW EXCLUSIONS OF AN IMPLIED WARRANTY, SO THIS DISCLAIMER MAY NOT APPLY TO YOU AND YOU MAY HAVE OTHER LEGAL RIGHTS THAT VARY FROM STATE TO STATE OR BY JURISDICTION.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL THE PRODUCER OF THIS SOFTWARE OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL THE PRODUCER OF THIS SOFTWARE BE LIABLE FOR ANY DAMAGES IN EXCESS OF THE LIST PRICE FOR A LICENSE TO USE BOOT−US, EVEN IF THE PRODUCER SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.



2.1.5 Distribution

If you like Boot−US, you are allowed to distribute it. The whole package Boot−US (non−licensed version) may be copied and distributed without restriction. The licensed version of Boot−US, especially the license file bootus.lic, is excluded from the distribution. The license file may not be distributed at all.

2.2 Registration

2.2.1 License types and prices

The following license types for Boot−US version 3.x.x can be purchased:

Amount	Version 3	Upgrade from V2	School license
1	18.50 EUR	7.50 EUR	
5	45.00 EUR	15.00 EUR	yes
20	90.00 EUR	25.00 EUR	yes
100	225.00 EUR	50.00 EUR	yes

These prices are final prices including tax and transportation. No further costs arise.

Owners of old V2 licenses may upgrade to a version 3 license for a reduced price (see above table).

Higher numbers or unlimited licenses are available on request from J−M−S and the author.

Public schools get the multiple licenses for half the price. According to the license type purchased the package Boot−US may be used only on the corresponding number of PCs. For example, a single PC license entitles the licensee to use Boot−US on one PC.

2.2.2 Obtaining licenses online

The registration service **J−M−S (http://www.j−m−s.com)** offers a convenient online registration of Boot−US. The following four links leads you **directly** to the appropriate web page at J−M−S (Boot−US has the numbers #0180 till #0183):

order Boot−US V3 (individual license)
order Boot−US V3 (5 licenses)
order Boot−US V3 (20 licenses)
order Boot−US V3 (100 licenses)

Upgrade licenses from version 2 to version 3 may be ordered by the following links from J−M−S (upgrade to Boot−US V3 has the numbers #4180 till #4183):

order upgrade from V2 to Boot−US V3 (individual license)
order upgrade from V2 to Boot−US V3 (5 licenses)
order upgrade from V2 to Boot−US V3 (20 licenses)



order upgrade from V2 to Boot–US V3 (100 licenses)

J–M–S offers a secure data transmission and accepts practically **any** way of payment (VISA, MASTERCARD, AMEX and DINERS, CHEQUES and many others). J–M–S has been provided with a license generator for Boot–US and will send the license file directly to the customer. This is the **easiest** and **fastest** way to obtain a license for Boot–US.

The file `english\payform.txt` in the installation directory of Boot–US contains an order form for J–M–S. If you cannot use the online registration (or if you do not want to use it) you can fill out this form and send it by e–mail, fax or postal letter to J–M–S.

We kindly ask you to make use of the registration service offered by J–M–S. You will **free** the author from pure administration duties and allow him to spend more time on the support and for improvements.

2.2.3 License file

When you purchase a license you will receive a license file together with detailed installation instructions (see next page Installing the license).



2.3 Installing the license

After purchasing a license you receive a license file `bootus.lic`. This license file can be used both for the GUI version and the command−line version of the configuration program Boot−US. Please **copy** the license file unchanged into the **installation directory** of Boot−US. In this directory you will find either the executable `bootus.exe` or `bootusc.exe`.
Remark: For the GUI version the default installation directory is `c:\program files\Boot-US`.

When you start the configuration program Boot−US (GUI or command−line) the next time the license file is recognized and all encrytped data (number of licenses, licensee, etc.) is extracted from the license file. When a valid license is recognized, the title of Boot−US (GUI) changes from "Boot−US [non−licensed version]" to "Boot−US", for example. All data extracted from the license file can be viewed by executing the menu command Help / About Boot−US....

When you install the boot manager the next time (or modify an existing installation of the boot manager) the extracted license is inserted into the boot manager.

Please note that the license is not inserted automatically into an already installed boot manager. You must re−install the boot manager. It is, however, not necessary to uninstall the boot manager. Just open the installation dialog and click "Next" on each page. All configuration data of the installed boot manager are already preset in the appropriate pages of the installation dialog.

The advantage of this approach is that the user controls all write access to the disk. Under no circumstance does Boot−US automatically change any data on the disk. All data written to the disk is generally controlled by the user.



2.4 Contact

In case of **orders**, **questions**, **suggestions for improvements** or **problems** contact us at the following addresses:

E–Mail: **ustraub@boot–us.com**

Postal address: Dr. Ulrich Straub
Grabenstrasse 71B
D–71116 Gärtringen
Germany

It is planned to fix errors as early as possible –– normally we expect a fixed version to be available within a few days. Improvements are expected to require larger modifications and hence more time. Therefore improvements will be available only in the following regular version. Questions from licensed users should be answered within a working day. Non–licensed users can expect an answer from within a day to up to a week.

When you report errors please describe the problem as detailed as possible. If necessary include screen shots and/or a complete trace file. Please send the files compressed.



3 Menu commands (GUI)

3.1 All menu commands of Boot–US

The following screenshot shows the configuration program Boot–US (GUI). This program runs under all WIN32/64 versions (Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8).

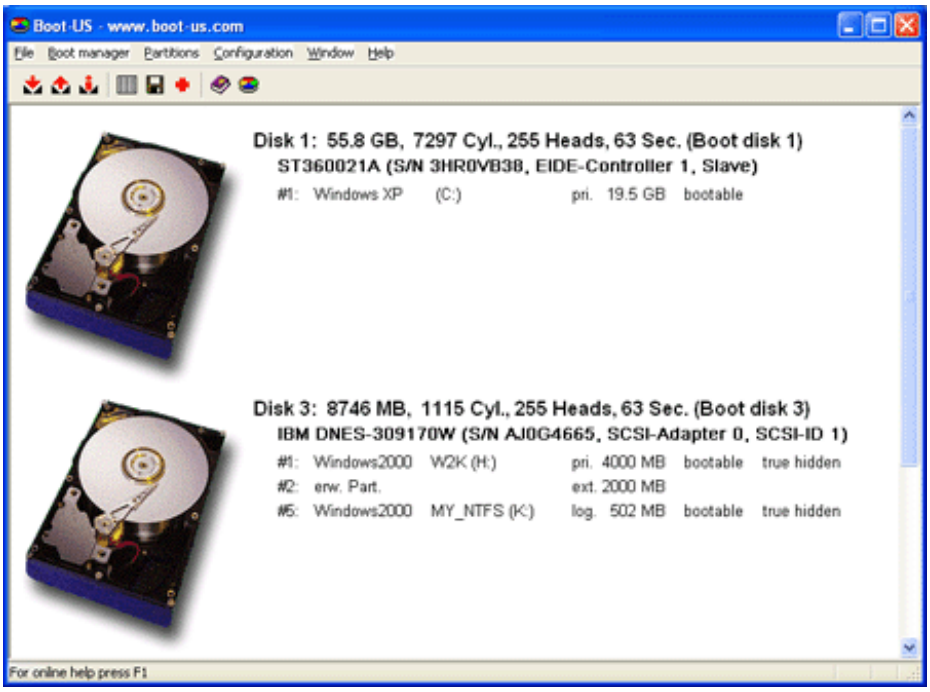


Fig. 4: Configuration program Boot–US (GUI)

The following summary lists all menu commands of the GUI version (fig 4) with a short description. An explicit description is found separately with each command.

File

Login...	Login into configuration program
Exit	Closes the configuration program

Boot manager

Install...	Installs boot manager
Remove...	Removes boot manager
Show info...	Shows info about installed boot manager

Partitions

Details...	Shows details of all partitions
Create...	Creates a new partition
Delete...	Deletes a partition
Save sectors...	Saves sectors to file
Restore sectors...	Restores sectors from file



Configuration

Base settings...	Specify basic parameters in INI file
Extended settings...	Specify extended parameters in INI file

Window

Refresh	Refreshes the display

Help

Contents...	Opens the online help
License...	Show license agreement in online help
Registration...	Show ordering information in online help
About Boot–US...	Show copyright, version and installed license



3.2 File / Login...

The menu command **File / Login...** allows to enter the password for the configuration program Boot−US. This menu command is only active when the password has been specified and the correct password has not yet been entered.

3.3 File / Exit

The menu command **File / Exit** closes the program Boot−US. This command does not modify any data and does not write anything to disk.



3.4 Boot manager / Install...

The menu command **Boot manager / Install...** opens a wizard to configure and install the boot manager. If the boot manager is already installed on the hard disk the data of this installed boot manager is used to initialize the wizard. Otherwise the wizard is initialized with standard values. The wizard consist of 6 pages to lead you through the configuration and installation of the boot manager.

Page 1: Partitions	Select partitions to be included in boot manager
Page 2: Password protection	Assign passwords to partitions
Page 3a: Startup options	Specify the startup options (part 1)
Page 3b: Startup options	Specify the startup options (part 2)
Page 4: Hide partitions	Select method for hiding of partitions
Page 5: Install target	Select installation target
Page 6: Installation	Show selected configuration and start installation

When you press the **Install** button on the last page, the boot manager is installed with the chosen configuration. In case of installation to the hard disk there are no further questions, and the installation begins immediately. When the boot manager is installed to the diskette, a request to insert a diskette into drive A: is issued.

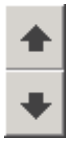
Comment:
When the boot manager is already installed on the hard disk and a second installation is done, the boot manager is removed automatically before it is installed again. This guarantees that the boot manager is installed just once.

3.4.1 Boot manager / Install / Partitions (page 1 of 6)

On the first page of the installation wizard (menu command **Boot manager / Install...**) you can select which partitions should be included in the boot manager. Only the bootable partitions are shown and can be selected.

Please note that the list of partitions includes also **booting from diskette**.

The name by which the partitions are displayed in the boot manager can be changed by the user (max. 12 char.). Also the description of the partition is editable (max. 33 characters). In order to change an entry doubleclick on it or press the **space bar** or the **F2** key. Additionally the name and the description can be changed by the context menu (right click).


The order of the entries in the boot manager can be changed by these buttons or by the context menu.


This button allows duplication of entries in the boot manager. By this means a partition can be booted with different settings. For example, one option would be to hide all partitions except the selected partition, while another option would be to make all partitions visible. Again the context menu also allows duplication of an entry.

3.4.2 Boot manager / Install / Password protection (page 2 of 6)

On the second page of the installation wizard (menu command **Boot manager / Install...**) you can specify the password protection in the boot manager.

Partitions in boot manager:
Here you can specify the password for all partitions in the boot manager. When a password has been defined for a partition then you need to enter the password when this partition is booted from the boot manager of Boot−US. Please **remember** the password carefully. There is no way in the boot manager to escape the password entry.

Administrator password for boot manager:
Additionally you can specify an administrator password for the boot manager. This password protects the configuration menu in the boot manager (**F10** key).

In order to specify a password you must enter the password twice. The new password is only accepted when both entries match. Please press the **RETURN** or **TAB** key after the password has been specified. You can abort the password entry by the **ESC** key.

The maximum length of a password is 20 characters. Starting with Boot−US 3.0.1 almost all printable characters of the 7 bit ASCII characters set are allowed. In order to avoid potential problems with the english keyboard driver in the boot manager you should use only the numbers `'0'-'9'` and letters `'a'-'x'` (upper and lower case). These characters occupy the same position on an english and german keyboard. When using special characters like `'!', '#', '-', '_', ...` on a non−english keyboard the boot manager maps the pressed key to the corresponding special character.

Comment:
In the non−licensed version of Boot−US you can specify passwords only for an installation of the boot manager to diskette. When you install the boot manager to hard disk and want to specify passwords you need a license.

Comment:
When you install the boot manager without password protection on a diskette you can boot the corresponding partitions by this emergency diskette even if you forgot the passwords.

3.4.3 Boot manager / Install / Startup options (page 3a of 6)

Startup options for the boot manager are configured on the third page of the installation wizard (menu command **Boot manager / Install...**). The startup options are distributed among two pages.

Start delay:
The start delay is set to zero seconds by default. The start of the boot manager can be delayed by up to a maximum of 10 seconds. This leaves the user more time to read the BIOS startup messages.

Available time:
The second time limit (max. 60 seconds) specifies the time available inside the boot manager to select a partition. When this time limit expires the currently selected partition is booted automatically. Inside the boot manager you can always stop this timer by pressing any key.

Unlimited:
You can also specify an unlimited second time limit. This would completely disable the second time limit.



Output start signal:
You can choose whether the boot manager should output a short acoustic signal upon startup or not. It is possible to change this setting also in the configuration menu of the installed boot manager.

Preselect last booted partition:
Here you can decide whether the boot manager starts up with the standard partition (first partition) preselected or the last booted partition. It is possible to change this setting also in the configuration menu of the installed boot manager.

3.4.4 Boot manager / Install / Startup options (page 3b of 6)

Startup options for the boot manager are configured on the third page of the installation wizard (menu command **Boot manager / Install...**). The startup options are distributed among two pages.

Visibility:
You can decide whether the boot manager should be visible or invisible when it is started. When the boot manager starts invisibly it is running in the background and the selected partition will be booted when the available time has counted down to 0.

Hotkey:
You can choose which key or key combination should bring an invisible boot manager to the foreground. Simple press the desired key or key combination. The following keys are allowed: `<F1>-<F12>`, `<A>-<Z>`, `<0>-<9>`, `<ESC>`, `<Enter>`, `<Space>`, `<Ins>`, `<Del>`, `<Home>`, `<End>`, `<PgUp>`, `<PgDown>` and the cursor keys. Most of these 61 keys can be combined with `<Ctrl>` and `<Alt>` or both. All together roughly 200 key combinations are possible.

Hint (30 character):
The hint text appears as soon as the boot manager is started and running in the background. Please note that the non−licensed version of Boot−US does not allow full control about this setting.

Show progress:
You can decide to show or disable a progress indicator displayed after the hint text. Please note that the non−licensed version of Boot−US does not allow full control about this setting.

3.4.5 Boot manager / Install / Hide partitions (page 4 of 6)

On the fourth page of the installation wizard (menu command **Boot manager / Install...**) you can specify the method for hiding partitions during the boot process.

(1) Do not change partition IDs:
The default is to hide no partition. In this case all partition IDs remain **unchanged**. All visible partitions remain visible and all hidden partitions remain hidden. When a partition is hidden it usually cannot be booted. This setting offers the greatest safety since no data in the MBR is changed during the boot process. However, the disadvantage is that potential conflicts with the assignment of drive letters may arise. For example, when Windows 95 and Windows 98 are installed independently on a disk only one of these partitions can get the drive letter C:. This C: partition may be booted without problems while the other partition will cause difficulties when booted.

(2) Hide primary partitions automatically:
The problems with the assignment of drive letters can be solved by hiding the non−selected partitions. If you decide to hide partitions automatically only the booted C: partition is visible, all other primary partitions working with drive letters will be hidden after the boot process. Hiding a partition means that the partition ID in the partition table (on the



hard disk) is modified such that it becomes an unknown partition. This has no consequence for the next boot process, i.e. the boot manager of Boot−US will boot a hidden partition without problems. In such a case the partition is made visible before it is booted, while the other partitions become hidden. In any case the booted partition is made visible while all other primary partitions are hidden.

(3) Hide pri. and log. partitions automatically:
Essentially the same as method (2), but now all logical partition are also hidden. Only the booted partition will be visible.

(4) Hide partitions individually (for experts):
This option is supplied for **experienced users** only. This options allows for each booted partition to define which partitions should be hid. You can hide primary and logical partitions. The boot manager hides exactly the specified partitions and unhides all other hideable partitions on all disks.

(5) TRUE hide primary partitions automatically:
(6) TRUE hide pri. and log. partitions automatically:
(7) TRUE hide partitions individually (for experts):
Essentially the same as the methods (2), (3) or (4). The only difference is that partitions are **true** hid. This means the boot sectors are additionally modified. True hiding is necessary when Windows 2000/XP/2003/Vista/2008/7/8 is involved, since the "simple" hidden partitions are still visible from Windows 2000/XP/2003/Vista/2008/7/8.

Comment:
Automatic partition hiding will be performed for **all** hideable primary and logical partitions on all disks referenced in the boot manager. Starting with Boot−US 3.0.0 the **highest** disk number referenced in the boot manager is determined. The automatic hiding does not act any more on the disks beyond the highest one, but only on the disks up to (and including) the highest disk.

Comment:
In case of the individual hiding the configuration program allows for any combination of partitions to be hidden. There is no test to determine if the selected combination can be successfully booted. **Your knowledge as an experienced user is needed**. For example, when Windows is installed twice, booting the "second" Windows requires that the "first" Windows is hidden, otherwise booting will fail.

Comment:
The boot manager hides or unhides only hideable partitions. All other partitions (e.g. Linux) are considered as non−hideable and consequently the partition ID of those partions remains unchanged in the boot process.

3.4.6 Boot manager / Install / Installation target (page 5 of 6)

On the fifth page of the installation wizard (menu command **Boot manager / Install...**) you can select the installation target of the boot manager. You can choose between installation in a separate primary partition, in the MBR, installation to a diskette or creation of an image file.

Primary partition on boot disk 1:
When the boot manager is installed on the **primary partition** an additional separate partition is created and the boot manager is installed on this partition. This partition can be used only by the boot manager. It does not contain a file system. This partition for the boot manager is created automatcially if it does not exist, otherwise an existing boot manager partition is reused. This install option is available only if the required primaray partition can be created, i.e. a free entry in the partition table and free space on the boot disk 1 is required. If one of these conditions is not met this option cannot be selected. This kind of installation has the big advantage that the boot manager partition is protected



against overwriting by third−party programs. This is not the case if the boot manager is installed in the MBR.

Option **permanent active:**
You also need to decide if the partition for the boot manager should be **permanent active** or not. In the first case the MBR remains unchanged except that an additional entry in the partition table is created and this partition is marked as active. The code in the standard MBR searches for the active partition and starts this partition. In this case it is our boot manager partition. which is active the whole time. The **advantage** of this type of installation is that the MBR remains unchanged. The boot manager can be disabled by any partition manager simply by deactivating the boot manager partition.
In the second case the MBR has to be changed to start the inactive boot manager partition. The boot manager partition can remain permanently inactive. As already mentioned the code in the MBR has to be modified. The **advantage** of this type of installation is that the booted partition can be set active. Especially for Windows 7 this is important since hibernation only works correctly if the Windows 7 partition is marked as active. In order to remove the boot manager a standard MBR has be restored. The uninstallation of the boot manager is therefore more elaborate than in the first case.

Remark: This option is automatically deactivated when no free primary partition is available for the boot manager (see above).

MBR on boot disk 1:
When the install target is set to **MBR** the boot manager will be written in the beginning of the track 0 of the first boot disk. This also changes the boot loader code in the MBR but leaves the partition table unchanged. Thus the boot manager will not disturb the system. The former MBR is copied completely to the second sector of the hard disk.
The **advantage** of this type of installation is that no entry in the partition table is consumed by the boot manager. It is always possible to install the boot manager to the MBR.
The **disadvantage** is that the boot loader of the MBR is changed. It may happen that some programs or operating systems consider the MBR as corrupt and try to "repair" it. However repairing the MBR removes the boot manager. The installation in the MBR represents a greater modification of the system than an installation in a separate primary partition. Additionally the main part of the boot maanger is not protected against overwriting by third−party programs.

Installation to diskette:
Installation to **diskette** allows a testing of the boot manager without any risk.
The **advantage** of this type of installation is that no data are changed on the hard disk. However, if automatic hiding of partitions is active then partitions will be hidden during the boot process. This means data in the partition table of the MBR need to be changed during the next boot process even if the boot manager resides on diskette. The **disadvantage** is that booting from this diskette is slow and inconvenient. Additionally the boot sequence has to be changed to something like "A: C:".

Create image file for CD installation:
This option will create an **image file** with the name "bmgrus.iso" or "bmgrus.img" in the installation directory of Boot−US. By default the image is created as a cd iso file. Burning this iso file to a cd automatically results in a **bootable cd** containing the configured boot manager. This should be easily possible with any cd writer program. If you decide to not create a cd iso image then the image file contains all sectors of a 1.44 MiB diskette with installed boot manager Boot−US. Again this image file can be used to create a bootable cd containing the boot manager Boot−US using appropriate cd burning tools, just a little more effort is required.

Recommendation:
For a **first test** the boot manager should be installed to **diskette** or **cd**. If booting from this diskette or cd works without problems one can think to install the boot manager to hard disk. Here one should prefer an installation to a **non−active primary partition**, since this type of installation could cause the least problems −− if at all. When this



partition is needed at some future time the boot manager can be easily "moved" to the MBR. All you need to do is run through the installation wizard and change the install target to MBR. All other data are already preset according to the installed boot manager.

3.4.7 Boot manager / Install / Installation (page 6 of 6)

On the last page of the installation wizard (menu command **Boot manager / Install...**) a summary of all selected boot manager settigns are displayed.

When you press the **Install** button on this last page, the boot manager is installed with these settings. In case of an installation to hard disk the installation will start immediately. However when the installation target is the diskette, a request to insert the diskette in drive A: will appear before the installation starts.



3.5 Boot manager / Remove...

The menu command **Boot manager / Remove...** is only enabled when the boot manager is installed on the first boot disk. When you click on this menu command a dialog is opened which allows removal of the boot manager from the hard disk. The boot manager is only removed after clicking the **Remove** button.

When the boot manager has been installed on the MBR the former boot loader code is re–written to the MBR during the removal process. This former boot loader code is read from the saved MBR, if present. Otherwise a functional standard boot loader is written to the MBR.

When the boot manager is installed on a primary partition this partition is deleted from the partition table on the first boot disk.

In both cases all **hidden** partitions are made visible on all disks. Additionally the partition which was active before the boot manager had been installed is made activate again. This information is also obtained from the saved MBR, if present. If necessary the first partition is activated. When more than one partition is active all but the first are deactivated. This guarantees that after the boot manager has been removed only one partition on the first hard disk is marked active.

After the boot manager is removed and the PC is restarted this active partition is booted.



3.6 Boot manager / Show info...

The menu command **Boot manager / Show info...** determines the data of an installed boot manager and displays them. Only a boot manager installed to the first hard disk will be checked, a boot manager installed to diskette is ignored.



3.7 Partitions / Details...

The menu command **Partitions / Details...** opens a dialog which shows a list of all partitions on all hard disks. All details like partition ID, partition type, beginning and end of partition, etc. are shown (see following table). The displayed data can be saved in a report (see below).

```
-------------------------------------------------------------------------
    #              - partition number
    Name           - selectable name of partition (max. 12 char.)
    Partition type - partition type (FAT16, FAT32, ...)
    Label          - drive letter and volume name
    ID             - ID of the partition (hex number)
    hidden         - is the partition hidden ?
    active         - is the partition marked as active ?
    pri/log        - type of partition (primary, extended, logical)
    bootable       - is the partition bootable ?
    Start CHS      - beginning of partition in CHS format
    End CHS        - end of partition in CHS format
    Start LBA      - beginning of partition in LBA format
    Num. Sec.      - number of sectors in partition
    Size           - size of partition
-------------------------------------------------------------------------
```

This dialog also allows the user to change the following three values:

- Partition name (max. 12 char.)
- Partitions can be activated or deactivated
- Partitions can be (simple/true) hid or made visible

These editable entries are displayed in another color.

All modifications will be stored until the dialog is closed by the **OK** button. Only then are the modfications written to disk.

Report...
Starting with version 2.1.0 the displayed partition data can be saved to a text file. After clicking on the report button you will be prompted to select the target file for storing the report. The format of the report is identical to the output of the report of the command–line version.

Comment:
In order to change a value please double click the corresponding entry or right click it to open the context menu or press the **space** or **F2** key.

Comment:
When a partition is **true** hid both the partition ID and the partition's boot sector are modified.

Attention:
When a partition is activated/deactivated or made hidden/visible then the corresponding entries in the partition table on the respective hard disk are changed. The result of these modifications become evident only at the next boot process. Only experienced people should perform these modifications, since the system might become unbootable if the entries are incorrectly changed.



3.8 Partitions / Create...

The menu command **Partitions / Create...** opens a wizard which guides you through all steps in creating a new partition. The wizard consists of the following pages:

Page 1: Select free space	Select the free space where the partition is to be created
Page 2: Partition size	Specify the size of the new partition
Page 3: Partition type	Specify the type of the new partition
Page 4: File system	Specify the file system for the new partition
Page 5: Create partition	Show parameters of new partition and create it

3.8.1 Partitions / Create / Select free space (Page 1 of 5)

On the first page of the wizard for creating partitions (menu command **Partitions / Create...**) you can select the free space where you want to create the partition.

You can also decide how the new partition should be aligned. For SSD drives the alignment of the new partition should be set to 1 MiB boundaries.

Please note that a logical partition can be created only within an extended partition. If you want to create a logical partition and no extended partition exists yet, you must first create the extended partition.

3.8.2 Partitions / Create / Size of partition (Page 2 of 5)

On the second page of the wizard for creating partitions (menu command **Partitions / Create...**) you can specify the partition size. The new partition can occupy the whole free space or you can leave some free space before and/or after the new partition.

You can enter the desired size either directly in the edit controls or by using the range control. The cursor keys allow changes to the size in small steps.

3.8.3 Partitions / Create / Partition type (Page 3 of 5)

On the third page of the wizard for creating partitions (menu command **Partitions / Create...**) you can select the partition type (primary/extended/logical partition).

In many cases the partition type is already determined by the position of the new partition. For example, a new partition within an extended partition is always a logical partition. When an extended partition is already present on the corresponding disk you cannot create another extended partition on it. You have the choice between primary and extended partition only when no extended partition exists on the corresponding disk . In all other cases the partition type is determined by the position of the partition.

3.8.4 Partitions / Create / File system (Page 4 of 5)

On the fourth page of the wizard for creating partitions (menu command **Partitions / Create...**) you can select the file system of the new partition.



The main file systems are selectable from a dropdown list. If you want to use a different file system you must enter the corresponding partition ID (in hex).

Additionally it is possible to format FAT12/FAT16/FAT32 partitions. In this case an empty FAT is always written. However you can choose whether the remaining sectors of the new partition are to be overwritten or not. When you select the **quick format** then only an empty FAT will be created, but the other sectors of the new partition will remain unchanged. When you do not select the quick formating then all sectors of the new partition will be changed. In this case an empty FAT will be created and all other sectors of the new partition will be overwitten with a fixed pattern.

3.8.5 Partitions / Create / Create partition (Page 5 of 5)

On the last page of the wizard for creating partitions (menu command **Partitions / Create...**) the parameters of the partition to be created are shown.

When you press the **Create** button and the new partition is to be formatted, then a confirmation of this operation will be requested. During formatting the sectors of the new partition will be changed irrecoverably. Only when no formatting is to be carried out the new partition will be created **without requesting further confirmation**. And only in this case the previous state can be recovered by restoring the partition table.

When the partition is formatted a progress dialog is shown. The first step is to create the new partition in the partition table. The second step is to create an empty FAT. Both tasks are finished within a few seconds. When the partition is to be completely formatted the third step is to overwrite all remaining sectors by zeros. This last task can be aborted at any time since the empty FAT has been created already.



3.9 Partitions / Delete...

The menu command **Partitions / Delete...** opens a wizard for deleting partitions. Only single partitions can be selected and deleted. The wizard consists of the following pages:

Page 1: Select partition	Select the partition to be deleted
Page 2: Delete method	Select the method for deleting the partition
Page 3: Delete partition	Show selected partition and delete it

3.9.1 Partitions / Delete / Select partition (Page 1 of 3)

On the first page of the wizard for deleting partitions (menu command **Partitions / Delete...**) you can select the partition to be deleted.

Only a single partition can be selected. In order to delete an extended partition you need to delete first all logical partitions in the corresponding extended partition, and then the extended partition.

3.9.2 Partitions / Delete / Method for deleting (Page 2 of 3)

On the second page of the wizard for deleting partitions (menu command **Partitions / Delete...**) you can specify the method for deleting the partition.

Quick delete:
In this case the partition is deleted only in the partition table. No sector within the partition is changed. As long as no other partition is created on the deleted disk space the deleted partition can be restored completely just by restoring the entry in the partition table.

Complete delete:
In this case the partition is deleted in the partition table and additionally all sectors of the partition are overwritten with zeros. When the partition has been deleted by this method the data within the partition are lost. By restoring the deleted entry in the partition table the deleted partition re–appears in the old size, but all sectors of the partition still contain only zeros and not the previous data.

3.9.3 Partitions / Delete / Delete partition (Page 3 of 3)

On the last page of the wizard for deleting partitions (menu command **Partitions / Delete...**) you can finally delete the selected partition.

When you press the **Delete** button a confirmation of this operation will be requested. When you have selected a complete delete of the partition all sectors of the partition to be deleted will be **unrecoverable**. When you have selected the quick delete method the partition will be deleted only from the partition table. Only in this case the previous state can be recovered by restoring the partition table.

When you have selected the complete delete method a progress dialog is shown while the sectors of the partition are overwritten. The overwriting can be aborted at any time since the partition has already been deleted from the partition table in the first step. When you abort the overwriting the partition is nevertheless deleted, but of course only the corresponding part of the partition has been overwritten.



3.10 Partitions / Save sectors...

The menu command **Partitions / Save sectors...** allows you to save all sectors which are modified during the operation of the configuration program Boot–US or the boot manager of Boot–US. A wizard is opened to guide you through the task of saving the sectors. Depending on the selected action only certain pages are displayed while others are skipped. The following list presents an overview over all pages of the wizard.

Select action	Select the desired action
Partition and boot sectors	Save all partition and boot sectors
Track 0	Save the track 0

3.10.1 Partitions / Save sectors / Select action (page 1)

On the first page of the wizard for saving sectors (menu command **Partitions / Save sectors...**) you can select the type of save operation.

Save partition and boot sectors:
This allows you to save all partition and boot sector from all disks to a single file. Additionally the whole track 0 of all disks is saved.

Save track 0 of boot disk 1:
This allows you to save only track 0 of the first boot disk. Other sectors will not be saved.

3.10.2 Partitions / Save sectors / Save partition and boot sectors (page 2 of 2)

On this page of the wizard for saving sectors (menu command **Partitions / Save sectors...**) you can save all partition and boot sectors from all disks to a single file. Additionally the complete track 0 of all disks is saved. This file contains both the raw sector data and also the disk position of each saved sector. Due to the additional data you need Boot–US to restore the sectors from this file.

Target file:
Here you can specify the file into which the partition and boot sectors will be saved.

Description (max. 80 char.):
Here you can specify a description with max. 80 characters. When restoring the sectors this description is displayed.

Comment: When the boot manager is installed the MBR and perhaps other sectors of the track 0 are modified. When partitions are true hid the boot sector of the respective partition is changed. All sectors which could be changed by the configuration program Boot–US or the boot manager of Boot–US are saved on this page.

When the boot manager is installed the original MBR is saved in the second sector of the track 0. In general the boot manager can reliably be removed by the menu command **Boot manager / Remove...** without affecting the remaining system.

In principle saving the partition and boot sectors to a file is therefore not necessary in order to remove the boot manager later. It is "only" a **precautionary** measure which should be taken before the boot manager is installed to the first boot disk.



Recommendation:
You should save a copy of the file with the partiton and boot sectors to **diskette**. If the file resides on disk it might not be accessible in case of problems.

3.10.3 Partitions / Save sectors / Save track 0 (page 2 of 2)

On this page of the wizard for saving sectors (menu command **Partitions / Save sectors...**) you can save all sectors of the track 0 of the first boot disk to a file. This file then contains the raw sector data without any modifications. This file could be restored to the track 0 by Boot–US. In case of problems, track 0 could be restored from this file with a standard disk editor or with the DOS command line tool **bootusc.exe**.

Target file:
Here you can specify the file into which track 0 is to be saved.

Recommendation:
You should save a copy of the file with the saved track 0 to **diskette**. If the file resides on disk it might not be accessible in case of problems.



3.11 Partitions / Restore sectors...

The menu command **Partitions / Restore sectors...** allows you to restore the previously saved sectors. A wizard is opened which guides you through the task of restoring the sectors. Depending on the selected action only certain pages are displayed whiles others are skipped. The following list presents an overview over all pages of the wizard.

Select action	Select the desired action
Partition and boot sectors	Restore partition and boot sectors
Track 0	Restore the track 0

3.11.1 Partitions / Restore sectors / Select action (page 1)

On the first page of the wizard for restoring sectors (menu command **Partitions / Restore sectors...**) you can select the type of restore operation.

Restore partition and boot sectors:
This allows you to restore all partition and boot sector of all disks from the specified file. Additionally the whole track 0 of all disks can be restored.

Restore track 0 of boot disk 1:
This allows you to restore track 0 of the first boot disk only. Other sectors will not be restored.

3.11.2 Partitions / Restore sectors / Restore partition and boot sectors (page 2 of 2)

On this page of the wizard for restoring sectors (menu command **Partitions / Restore sectors...**) you can restore all partition and boot sectors from all disks from the selected file. Additionally the complete track 0 of all disks can be restored. The original state of the sectors will be restored. All modifications done after the partition and boot sectors were saved will be reversed.

File:
Here you can specify the file from where the partition and boot sectors are to be restored.

When the specified file is opened the internal format of the file is checked. When the format is correct, the description and the list of saved sectors is displayed. If the file format is not correct, the display is empty and restoration is not possible.

If a correct file format is recognized the saved sectors are compared with current sector content on the disks. The modified sectors are marked. Initially only the modified sectors are selected for restoration. You can extend this selection or remove sectors from the selection. The total number of selected sectors is dispayed.

When you click on **Restore** the selected sectors will be restored from the specified file.

3.11.3 Partitions / Restore sectors / Restore track 0 (page 2 of 2)

On this page of the wizard for restoring sectors (menu command **Partitions / Restore sectors...**) you can restore all sectors of track 0 of the first boot disk from the specified file. This file contain the raw sector data without any modifications. When you restore track 0 the complete partition table is also restored. The sector data are completely copied from the specified file to track 0 of the first boot disk. This restores the original state of track 0. All



modifications done after track 0 has been saved will be recovered.

Before the file is written to track 0 it is verified that the file size agrees with the size of track 0. The contents of the file is essentially not checked. It only verifies that the partition sector signature is present in the first sector of the file.

File:
Here you can specify the file containing the sectors of track 0.



3.12 Configuration / Basic settings...

The menu command **Configuration / Basic settings...** allows to set the standard options. This comprises all options which the user modifies during normal operation. All options are stored in the INI file `bootus.ini` in the installation directory of Boot–US. The only exception is the password protection which offers additional storage locations. The INI file is read on every start and each time the options are changed.

The following options can be set:

Language	Select language
Password	Password protection for configuration program
Boot disk	Specify order of boot disks

3.12.1 Configuration / Basic settings / Language

On this page you can explicitly specify which language should be used inside Boot–US and the boot manager.

German
All screen outputs e.g. menu commands, dialogs and all message boxes are displayed in German. Also the online help is displayed in German. Only the trace messages are still created in english.

English
All screen output and the online help is displayed in English. The files `english\bootus_eng.dll` containing the English resources and `english\bootus_eng.chm` containing the English online help must reside in the installation directory of Boot–US.

3.12.2 Configuration / Basic settings / Password

On this page the **password for the configuration program Boot–US** can be specified. When a password is set the configuration program Boot–US can be started only after entering the correct password.

Password for Boot–US and **Confirmation**
These two fields allow you to enter the password and the password confimation.

In order to specify a password you must enter the password twice. The new password is only accepted when both entries match. Please press the **RETURN** or **TAB** key after each password has been specified. You can abort the password entry by the **ESC** key.

The maximum length of a password is 20 characters. Valid characters are almost all printable characters of the 7 bit ASCII character set. In order to avoid potential problems with the english keyboard driver in the boot manager you should use only the numbers `'0'-'9'` and letters `'a'-'x'` (upper and lower case). These characters occupy the same position on an english and german keyboard. When using special characters like `'!', '#', '-', '_', ...` on a non–english keyboard the boot manager maps the pressed key to the corresponding special character.

Store password in boot manager
You can select where the password should be stored. The **safest possibility** is to store the password within the boot manager. This practically prevents any manipulation of the password. Only by deinstalling the boot manager or by changing the password by the configuration program can the password be modified or removed. This way of storing



the password is only possible when you have purchased a **license** for Boot–US and when the boot manager of version 1.3 (or higher) is already installed.

Store passwort in registry
In this case the password is stored encoded in the registry under the key
`HKEY_LOCAL_MACHINE\Software\ustraub\Boot-US`
Please note that this key is **user independent**. This means a second installation of Boot–US under another user will access the same key and thus the second installation is also password protected. This way of storing the password does not require a license or a boot manager which is already installed. However the password protection can be easily removed by **deleting the registry key**.

Store passwort in INI file
Here the password is stored encrypted in the INI file in the installation directory. This way of storing the password offers the least security and can be overcome easily, e.g. by deleting the password in the INI file, or by deleting the whole INI file, or by re–installing Boot–US into another directory.

Priority of storage locations
When the configuration program is started it searches for a password in the following order:

- Boot manager (highest priority)
- Registry
- INI file (lowest priority)

As soon as a non–empty password is found this password is used. It is only when no password has been found in any location that the configuration program starts without password protection.

Overcome the password protection
It is clear that the password can be changed or removed by the configuration program. In case you have forgotten the password and can not start the configuration program, you can overcome the password protection by the following measure:

- Manually deinstall the boot manager
- Delete the password entry in the registry
- Delete the password entry in the INI file

Comment:
Please do not mix up the password for the configuration program with the administrator password for the boot manager. Both passwords are completely **independent**. The administrator password of the boot manager is used only inside the boot manager. The password for the configuration program is is used only to protect the start of the configuration program.

3.12.3 Configuration / Basic settings / Boot disk

On this page the order of the **hard disks upon booting** is specified. This order could be different from the apparent disk order under Windows, especially when both IDE and SCSI disks are present. The relative order of IDE and SCSI disks is determined by the BIOS. However, under Windows the disks usually appear in a fixed order.

Please note that it is not possible to change the physical order of the disks in this dialog. The order of the hard disks can be partially modified in the BIOS. The order of IDE disks is usually determined by connecting the disk to the



primary or secondary IDE channel and by configuring the disk as master or slave. The order of SCSI disks is usually determined by the SCSI ID.

Comment:
In general the boot manager of Boot−US is installed on the disk specified as boot disk 1. Besides the boot manager of Boot−US uses only the boot disk numbers and not the "apparent" Windows disk numbers.

Comment:
Before you install the boot manager you can specify an order of the boot disks which is can be different than the current order of the boot disks. The boot manager is then installed with these **future** boot disk numbers. Of course, after the installation of the boot manager you must change the actual order of the boot disks to reflect the order used in the installed boot manager.



3.13 Configuration / Extended settings...

The menu command **Configuration / Extended settings...** allows setting of non–standard options. This comprises all options which one never or seldom modifies during normal operations. All options are stored in the INI file `bootus.ini` in the installation directory of Boot–US. The INI file is read on every start and each time the options are changed.

The following options can be set:

Trace file	Switch tracing on/off
Startup tip	Enable/disable startup tip
Backup	Set backup options
Warnings	Enable/disable CHS warnings
Units	Select SI/IEC units
Alignment	Alignment of partitions

3.13.1 Configuration / Extended settings / Trace

On this page tracing can be switched on or off. The trace file contains a protocol of the main function calls. For example when Boot–US is started all partitions found with all details are written to the trace file. Please note, that tracing should be activated only for **error determination**, since the program is slowed down noticeably by using tracing.

3.13.2 Configuration / Extended settings / Startup tip

On this page the startup tip can be activated or deactivated.

3.13.3 Configuration / Extended settings / Backup

On this page the backup options can be specified. All partition and boot sectors on all disks are saved automatically immediately before the following operations are performed::

- Installing the boot manager
- Hiding/activating partitions
- Restoring sectors

Backup directory:
Here you can specify the backup directory. This directory must exist already, it is not created here. All automatically created backups are stored in the backup directory. The backup directory is also the initial target directory for manual backups, however, the target directory can be changed.

Max. number of files:
The max. number of files specifies how many automatic backups should be stored (0–99). The default is 20, i.e. the files `'auto01'...'auto20'` will store the automatic backups. When the maximum number of files is reached the oldest file is overwritten. When you specify 0 as max. number of files the automatic backup is **disabled** completely. Already saved files will not be deleted by selecting 0 as max. number of files.



3.13.4 Configuration / Extended settings / Warnings

On this page potential warnings about invalid CHS values in the partition tables can be disabled or enabled. The partition tables are not changed at all, only the display of the warning message is suppressed.

Techn. background:

When partitions exceed the 8 GB limit the CHS values in the partition tables are not sufficient for specifying the beginning or end of the partitions. Only the 32−bit LBA values in the partitions tables are valid. The 24−bit CHS values in the partitions values can contain only "truncated" values. In order to mark the CHS values as invalid the common practice is to set the cylinder number to the highest possible value (1023). Some partition managers use the value 1022 instead of 1023 for marking invalid CHS values.

When the partitions exceed the 8 GB limit and the cylinder number is not 1023 or 1022, Boot−US will display a warning about invalid CHS values.

The appearance of this warning means that the corresponding partition table entry has been created with a partition manager **which does not comply with established standards**, i.e. it does not set the cylinder number to 1023 (or 1022) for invalid CHS values. In such a case you should use only modern partition managers for accessing the partition tables. In other words the partition manager should work internally with the LBA values and should have no problems with the 8 GB limit or large hard disks.

3.13.5 Configuration / Extended settings / Units

Boot−US can display disk and partition sizes either in units of KB, MB, GB, TB or alternatively in units of KiB, MiB, GiB, TiB.

The standard SI units KB, MB, GB and TB are powers of 1000, for example 1 GB = 1000^3 Byte = 10^9 Byte.

On the other hand the units KiB, MiB, GiB and TiB are specified as powers of 1024, for example 1 GiB = 1024^3 Byte = 2^{30} Byte = 1.073 GB. These units were standardized by IEC in 1999. Starting with Boot−US 3.0.1 these IEC units are used by default.

3.13.6 Configuration / Extended settings / Alignment

On this page you can specify which convention for aligning the partition is expected by Boot−US.

Alignment on 1 MiB boundary:
This allows you to set the alignment on 1 MiB boundaries. If this option is not selected then the old alignment on cylinder boundaries is used.

The actual alignment of already existing partitions remains unchanged by this choice. The selected alignment will be used however when creating new partitions in Boot−US (GUI). If the display of warnings is activated then you can easily verify whether partitions are aligned according to the new or old convention.

Techn. background:

Starting with Windows Vista and Windows 7 a new convention for aligning partitions was established, namely aligning partitions on 1 MiB boundaries. Especially for SSD drives the new alignment allows faster access to sectors within partitions. If partitions were aligned according to the old cylinder boundaries on SSD drives the sectors reside on "odd" addresses and such positions cannot be accessed with the highest speed. For traditional magnetic drives



however the alignment of partitions should not have a large effect on the performance.



3.14 Window / Refresh

The menu command **Window / Refresh** rebuilds the complete partition list and refreshes the display. In general this is done automatically when necessary. A manual refresh of the display is normally not required.

After a refresh of the display the information shown is the same as that immediately after a restart of Boot–US.



3.15 Help / Contents...

The menu command **Help / Contents...** opens the online help.

Comment:
In any dialog you can obtain context−sensitive help by pressing the F1 key.

3.16 Help / License...

The menu command **Help / License...** opens the online help und shows the license agreement. Additionally this page contains the disclaimer of warranty and the conditions for the redistribution of Boot−US.

3.17 Help / Registration...

The menu command **Help / Registration...** opens the online help and shows informations about ordering of licenses.

3.18 Help / About Boot−US...

The menu command **Help / About Boot−US...** shows the copyright, the version of Boot−US and the current license type and licensee.





4 Command–line version

4.1 Overview

The command–line program (fig. 5) ––– program name `bootusc.exe` ––– is used to configure, install and remove the boot manager Boot–US. Additionally the command–line program allows you to hide and activate partitions and it allows you to save all partition and boot sectors. In short the command–line version of Boot–US supports essentially the **same set of operations** as the GUI version.

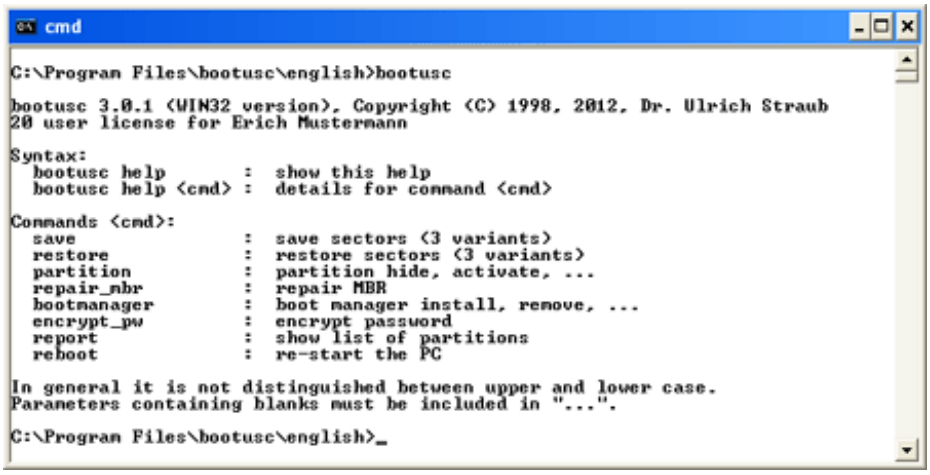


Fig. 5: Configuration program Boot–US (command–line version)

The command–line program exists in two versions, a DOS version and a WIN32 version. The DOS version can be executed only under DOS (or in DOS mode), the WIN32 version can be executed in a command–line window under any 32–bit or 64–bit Windows version (Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8).

The language of the command–line program is fixed to either German or English. Contrary to the GUI version of Boot–US the language of the command line program cannot be switched.

The WIN32 version is intended for companies with a large number of PCs. Almost all functions in the WIN32 version require a 20 user license (or higher). Only the report function does not require a license.

The DOS version is intended for private usage. Furthermore the DOS version allows you to use the functions of the package Boot–US even if Windows cannot be accessed. The DOS version is subject to the same license requirements and restrictions as the GUI version of Boot–US. For example, the installation of the boot manager to diskette does not require a license, while the installation to disk is available only in the licensed version.

Comment:
Creating and deleting a partition is currently only supported by the GUI version.

Comment:
It sometimes happens that users first look at the command–line version when they have their first contact with the package Boot–US. In such a case it is recommended that you switch to the GUI version of the configuration program. The functions offered by the package Boot–US are easily accessible in the GUI version. For a first contact with Boot–US the GUI version is much more suitable than the command–line version.



4.2 Installation

The command–line program of Boot–US is distributed as a standard ZIP file. This ZIP file contains both the DOS and the WIN32 version of the command–line program. The ZIP file can be un–zipped in any Windows shell (32 bit and 64 bit). After un–zipping you should have the following files in the unzip directory:

File name	Comment
readme.txt	help for command–line version
bootusc.ini	example INI file for bootusc.exe
bmgrus.ini	example setup data for boot manager
dos32\bootusc.exe	the command–line program for DOS
win32\bootusc.exe	the command–line program for WIN32/64
win32\disk32.dll	direct disk access for Windows 95/98/ME (32–bit part)
win32\disk16.dll	direct disk access for Windows 95/98/ME (16–bit part)

The DOS and WIN32 version of `bootusc.exe` is contained in the subdirectory `dos32` respectively `win32`. The sample ini files for the DOS and WIN32 version are identical.

It is recommended that you install the DOS version of `bootusc.exe` to a bootable DOS diskette so that you can access the program `bootusc.exe` in case of problems.

Since version 2.1.5 the DOS command–line version runs in protected mode. This is indicated by the term "DOS32". This change means that the program can use the whole available memory. Thus the DOS version is no longer restricted by the 640 KB barrier.



4.3 INI file bootusc.ini for command–line program

The INI file for the command–line version `bootusc.exe` has the name `bootusc.ini` and must reside in the same directory as the executable `bootusc.exe`. The package contains a sample INI file.

By default the command–line program does **not** require an INI file. The INI file is necessary only for specific tasks. When an INI file for `bootusc.exe` is present it is read each time the command–line program is started.

The format of the INI file is identical to the Windows standard format of INI files, i.e. there are sections with keywords. All section names are different and within a section all keywords are different. However the same keyword may appear in different sections.

4.3.1 Trace

The trace function of the program `bootusc.exe` can be enabled in the INI file. The following example shows the corresponding entries in the INI file:

```
[TRACE]
trace=1                         # default: 0
tracefile=bootusc.trc           # default: bootusc.trc
```

By default tracing is disabled.

In the example above the tracing is enabled. The trace is written to the default trace file `bootusc.trc` or the specified trace file.

4.3.2 Backup

Starting with version 1.5.4 an automatic backup of partition and boot sectors can be enabled before the boot manager is installed. This backup allows an easy restore of the previous contents of all partition and boot sectors.

The following example shows the corresponding entries in the INI file:

```
[BACKUP]
NumFiles=20                     # Default: 0
Directory=C:\temp\backup        # directory to store the backup files
LastFile=1                      # do not manually edit the parameter 'LastFile'
                                # it is automatically incremented by bootusc.exe
```

By default the backup of partition and boot sectors is disabled. In order to enable the automatic backup the parameter `NumFiles` must be set to a positive value, for example `NumFiles=20`. When the backup is enabled, the partition and boot sectors from all disks are saved before the boot manager is installed. The backup file is created in the specified directory. The file name is `auto<nn>.sec`, where `<nn>` is a number between 1 and `NumFiles`. The file number `<nn>` is automatically incremented before each backup. It is reset to 1 when the maximum number `NumFiles` is exceeded. The parameter `LastFile=<nn>` stores the number `<nn>` used in the last backup operation. The value of this parameter is automatically incremented. The keyword `LastFile` should not be created or modified by the user.



4.3.3 Password for configuration program

Starting with version 1.5.4 the execution of the configuration program Boot−US (GUI and command−line) is protected by a password. Both programs (GUI and command−line) are protected by the same password. When a password is specified this password must be provided in order to use the configuration program.

Typically this password would be stored within the boot manager. As long as the boot manager is installed the password is then required to execute the configuration program (GUI and command−line). When the configuration program is protected by a password, the default behaviour is to prompt the user to enter the password interactively.

However the command−line program allows to run most commands also in batch mode (non−interactively). In batch mode the user should not be prompted. In order to achieve this behaviour when the password protection has been activated the password can be specified in encrypted form in the INI file `bootusc.ini`. When the command−line program finds the correct password in the INI file `bootusc.ini` the user is not prompted any more to enter the password. This feature allows to write scripts which execute all commands without any user interaction.

The following example shows the corresponding entries in the INI file for specifying the password:

```
[PASSWORD]
Password=<xxx>              # encrypted Boot-US password
                            # default: no password
```

The keyword `Password` specifies the password for the configuration program. It is necessary to specify this password in encrypted form.

The encrypted password can be determined by the command `bootusc encrypt_pw`, see below.

4.3.4 Warnings

Starting with version 1.6.2 many integrity checks of the partitions are performed. As a result of these checks error and warning messages could be created and displayed with the corresponding partition.

In general error messages indicate a severe problem. Error messages should not be ignored. Error messages canot be suppressed.

On the other hand warning messages typically indicate "only" a violation of established standards. Usually these warnings can be ignored. Typical examples of warning messages are the violation the convention that partitions should start/end on cylinder boundaries or 1 MiB boundaries or non−standard CHS values in the partition table when the partition exceeds the 8 GB limit.

The warnings can be suppressed by the following INI parameter:

```
[STARTUP]
ShowWarnings = 0            # default: 1
```

When `ShowWarnings` is set to 0 (null) no warnings are displayed.



4.3.5 Use IEC units

Boot–US can display disk and partition sizes either in units of KB, MB, GB, TB or alternatively in units of KiB, MiB, GiB, TiB.

The standard SI units KB, MB, GB and TB are powers of 1000, for example 1 GB = 1000^3 Byte = 10^9 Byte.

On the other hand the units KiB, MiB, GiB and TiB are specified as powers of 1024, for example 1 GiB = 1024^3 Byte = 2^{30} Byte = 1.073 GB. These units were standardized by IEC in 1999. Starting with Boot–US 3.0.1 these IEC units are used by default.

The IEC units are activated by the following INI parameter:

```
[STARTUP]
UseIECUnits = 1             # default: 1
```

4.3.6 Individual license texts in boot manager

The 100 pc license (or higher) allows you to change the license displayed in the boot manager. This is achieved by the following setting:

```
[LICENSE]
Text1=my new license text1
Text2=my new license text2
```

Note:
You can add this section also to the INI file `bootus.ini` in order to use the changed license texts when installing the boot manager by the Boot–US (GUI).



4.4 Online help

The command–line program `bootusc.exe` contains a short online help for all supported commands. This help is invoked by

```
bootusc help
```

You can get extended online help for a specific command `<cmd>` by

```
bootusc help <cmd>
```

When a command `<cmd>` has different variants `<subcmd>` you get the online help by using

```
bootusc help <cmd> <subcmd>
```



4.5 Operations on sector level

The following commands can be applied to single or groups of sectors:

- Save track 0
- Save sector
- Save partition and boot sectors
- Restore track 0
- Restore sector
- Restore partition and boot sectors

4.5.1 Save track 0

The command for saving the complete track 0 of boot disk 1 to a file is:

```
bootusc save track0 [file=<filename>]

Examples:
bootusc save track0
bootusc save track0 file=c:\temp\track0.bin
```

The default file name is `track0.bin` in the current directory. This file is used when no file name is specified. Otherwise the track 0 is saved to the specified file `<filename>`.

Comment:
The above command is identical to the corresponding command of the GUI version. The created file containing the track 0 can be restored to the boot disk 1 both by the command–line version and the GUI version of Boot–US.

4.5.2 Save sectors

The command for saving arbitrary sectors of a certain disk to a file is:

```
bootusc save sector {chs=c/h/s | lba=x} [num=n] [drive=d] file=<filename>

  c/h/s = cylinder / head / sector number
  x     = LBA number
  n     = total number of sectors (1..127)
  d     = disk d (1..127) or diskette (A,B)

Examples:
bootusc save sector LBA=0 num=1 drive=A file=c:\temp\mbr.bin
bootusc save sector CHS=0/0/1   drive=2 file=c:\temp\mbr.bin
```

It is necessary to specify the sector(s) to be saved in the CHS or LBA format. Additionally the file name must be specified where the sector data are to be stored.

By default the command for saving sectors refers only to a single sector on the first disk. If required the command can be extended to multiple sectors by the option `num`. Additionally the option `drive` allows to access sectors on other disks.



The file where the binary sector data are stored contains only the plain sector data. This file does not contain any position information. In addition to using Boot−US to restore the file to disk any disk editor can be used as well.

4.5.3 Save partition and boot sectors

The command for saving all partition and boot sectors on all disks into a single file is:

```
bootusc save partsec file=<filename>

Example:
bootusc save partsec file=c:\temp\backup.sec
```

It is necessary to specify the file name where the sector data are to be stored. This target file contains the binary sector data together with disk positition information for each data block. Due to the additional sector data only Boot−US can be used for restoring the file to disk.

Comment:
The above command is identical to the corresponding command of the GUI version. The created file containing the partition and boot sectors can be restored both by the command−line version and the GUI version of Boot−US.

4.5.4 Restore track 0

The following command restores the complete track 0 of the boot disk 1 from a file:

```
bootusc restore track0 [file=<filename>]

Examples:
bootusc restore track0
bootusc restore track0 file=c:\temp\track0.bin
```

The default file name is `track0.bin` in the current directory. This file is used when no file name is specified. Otherwise track 0 is restored from the specified file `<filename>`.

Comment:
The above command is identical to the corresponding command of the GUI version. The file containing the track 0 can be restored to the boot disk 1 both by the command−line version and the GUI version of Boot−US.

4.5.5 Restore sectors

The command for restoring arbitrary sectors of a certain disk from a file is:

```
bootusc restore sector {chs=c/h/s | lba=x} [num=n] [drive=d] file=<filename>

  c/h/s = cylinder / head / sector number
  x     = LBA number
  n     = total number of sectors (1..127)
  d     = disk d (1..127) or diskette (A,B)

Examples:
bootusc restore sector LBA=0 num=1 drive=A file=c:\temp\mbr.bin
bootusc restore sector CHS=0/0/1   drive=2 file=c:\temp\mbr.bin
```



It is necessary to specify the sector(s) to be restored in the CHS or LBA format. Additionally the file name must be specified where the sector data are to be read from.

By default the command for restoring sectors refers only to a single sector on the first disk. If required the command can be extended to multiple sectors by the option `num`. Additionally the option `drive` allows to access sectors on other disks.

The file where the binary sector data are stored contains only the plain sector data. This file does not contain any position information. Additionally to using Boot−US for restoring the file to disk any disk editor can be used as well.

4.5.6 Restore partition and boot sectors

The command for restoring all partition and boot sectors on all disks from a single file is:

```
bootusc restore partsec file=<filename>

Example:
bootusc restore partsec file=c:\temp\backup.sec
```

It is necessary to specify the file name where the sector data are to be read from. This file contains the binary sector data together with disk positition information for each data block. Due to the additional sector data only Boot−US can be used for restoring the file to disk.

Comment:
The above command is identical to the corresponding command of the GUI version. The file containing the partition and boot sectors can be restored both by the command−line version and the GUI version of Boot−US.



4.6 Operations on partition level

The following commands can be applied to whole partitions:

- Hide a partition
- Unhide partitions
- Activate a partition
- Deactivate partitions

4.6.1 Hide partition

The command for hiding a partition is:

```
bootusc partition hide {label=<label> | partnr=n | lba=x}
                       [true] [drive=d] [batch]
  d = disk 1..127
  n = partition number (see report)
  x = LBA number of partition (see report)

Examples:
bootusc partition hide drive=1 label=WIN_95 batch
bootusc partition hide drive=1 label="WIN 95"
bootusc partition hide partnr=4
bootusc partition hide true LBA=63
```

The specified partition on the disk `d` will be hidden. When `true` is specified the partition will be true hid. The partition to be hidden must be identified either by the volume label `<label>`, the partition number or the LBA number `x` of the partition begin. This information is displayed by the report function. When no drive is specified disk 1 is assumed. When `batch` is specified no user intervention is necessary.

4.6.2 Unhide partitions

The command for unhiding partitions is:

```
bootusc partition unhide [drive=d] [batch]

  d = disk 1..127

Examples:
bootusc partition unhide
bootusc partition unhide drive=2
```

The unhiding of partitions is carried out for all partitions on the first respectively on the specified disk `d`. This is the only modification in the partition table of the MBR of the respective disk. The boot loader code in the MBR remains unchanged completely. When no disk is specified the partitions on disk 1 are unhid. When `batch` is specified no user intervention is necessary.

Starting with bootusc 1.5.0 true hidden partitions are also unhidden by this command. In order to unhide a true hidden partition the boot sector of the respective partition must be reset to the previous state.



When you have activated the automatic hiding of C: partitions (or the individual hiding of partitions) during the installation of the boot manager of Boot–US, on every boot process all primary (and logical) non–selected partitions (or the specified partitions) will be hidden. This happens also when you have performed an installation of the boot manager on a diskette. There are configurations where several partitions are involved in a successful boot process. When one of these partitions is hidden, the boot process may fail. This boot problem can be easily solved by unhiding all partitions.

4.6.3 Activate partition

The command for activating a partition is:

```
bootusc partition activate {label=<label> | partnr=n | lba=x}
                           [drive=d] [batch]
  d = disk 1..127
  n = partition number (see report)
  x = LBA number of partition (see report)

Examples:
bootusc partition activate drive=1 label=WIN_95 batch
bootusc partition activate drive=1 label="WIN 95"
bootusc partition activate partnr=4
```

The specified primary partition on the disk `d` is activated. Only primary partitions can be activated. Partitions must be activated individually, i.e. the respective partition must be specified either by the volume label `<label>`, the partition number `n` or the LBA number `x` of the partition begin. All this information is displayed by the report function. When a certain partition is activated all other primary partitions on the same disk are deactivated automatically. When no disk is specified disk 1 is assumed. When `batch` is specified no user intervention is necessary.

4.6.4 Deactivate partitions

The command for deactivating all partition on a certain disk is:

```
bootusc partition deactivate [drive=d] [batch]

  d = disk 1..127

Examples:
bootusc partition deactivate drive=1 batch
bootusc partition deactivate drive=2
```

This deactivates all partitions on the specified disk `d`. No partition remains active on this disk. When no disk is specified disk 1 is assumed. When `batch` is specified no user intervention is necessary.

A configuration where no partition is active will cause problems when trying to boot without a boot manager, since the MBR code expects to find exactly one active primary partition. If no active partition is found, the MBR code stops the boot process and displays an error message.

This command was requested by a customer in order to avoid some problem with the assignment of drive letters. This command might not be useful for other customers.



4.7 Operations for boot manager

The following commands are available for the administration of the boot manager:

- Install boot manager
- Uninstall boot manager
- Show status about installed boot manager
- Update entry in boot manager
- Preselect other entry in boot manager

Comment:
The commands listed above are compatible with the respective commands of the GUI version of the configuration program. Updating the entries of an already installed boot manager is available only in the command–line version.

4.7.1 Install boot manager

The command for installing the boot manager is:

```
bootusc bootmanager install [file=<filename>] [batch]

Examples:
bootusc bootmanager install
bootusc bootmanager install file=c:\temp\bmgrus.ini batch
```

The configuration parameters of the boot manager are read from a separate configuration file specified by the parameter `file`. The default configuration file is called `bmgrus.ini` residing in the same directory as the command–line program `bootusc.exe`. When `batch` is specified no user intervention is necessary.

4.7.1.1 Contents of bmgrus.ini

The configuration file `bmgrus.ini` contains a global section [GLOBAL] and separate sections [ENTRY_<nn>] (<nn> = 1..N) for every entry in the boot manager.

General rules:
The sections must be specified in upper case. The keywords can be specified in any case. The symbol <nn> denotes an integer, <nn> denotes a string.

4.7.1.2 Keywords in section [GLOBAL]

```
INSTALL_TARGET = PRIMARY | PRIMARY_INACTIVE | MBR | DISKETTE | IMAGE | IMAGE_CDISO
                                    # Install target of boot manager.
                                    # Default is PRIMARY_INACTIVE.

STARTUP_DELAY = <nn>                # Startup delay in sec.
                                    # Default is 0 (sec).

AVAILABLE_TIME = <nn> | UNLIMITED   # Time available for selecting another partition.
                                    # Default is 30 (sec).

START_SIGNAL = YES | NO             # Output start signal upon startup of boot manager ?
                                    # Default is NO (no start signal).
```



```
SELECT_LAST_BOOT = YES | NO          # Should the top partition or the last booted
                                     # partition be selected ?
                                     # Default is NO (top partition is selected).

INVISIBLE_START = YES | NO           # Invisible start of  boot manager ?
                                     # Default is NO (boot manager is visible).

INVISIBLE_HOTKEY = <hotkey>          # Hotkey for displaying invisible boot manager.
                                     # Mandatory if the boot manager is invisible.
                                     # Example: <ALT> + <B>

INVISIBLE_STARTTEXT = <text>         # Text (30 ch.) displayed while boot manager is invisible.
                                     # Default is "Booting".

INVISIBLE_PROGRESS = YES | NO        # Show progress while boot manager is invisible ?
                                     # Default is YES (show progress).

SIMPLE_CFG_MENU = YES | NO           # Should only the simple commands in the
                                     # configuration menu be displayed ?
                                     # Default is NO (show all commands).
                                     # When YES is specified the two commands
                                     # "Unhide all partitions" and
                                     # "Uninstall boot manager" are not shown.

DISABLE_HIDE_TEXT = YES | NO         # Should the hide state be displayed in the boot manager?
                                     # Default is NO (show hide state).
                                     # When YES is specified no hide state is displayed.

ENCRYPT_PASSWORDS = YES | NO         # Specified whether all passwords specified
                                     # in this configuration file are encrypted
                                     # or plain text.
                                     # Default is NO meaning passwords are not encrypted.

BOOTUS_PASSWORD = <xxx>              # Admin password of configuration program Boot-US
                                     # (encrpyted or plain text).
                                     # The specified password is stored in the boot manager.
                                     # It protects the command line and GUI version of Boot-US.
                                     # Accepted only in licensed version.
                                     # Default is empty string.

BMGRUS_PASSWORD = <xxx>              # Admin password of boot manager (encrpyted or plain text).
                                     # The specified password protects the configuration menu
                                     # of the boot manager.
                                     # Accepted only in licensed version.
                                     # Default is empty string.

KEYBOARD_LAYOUT = ENGLISH | GERMAN   # Keyboard layout: english or german
                                     # The layout is required only if special characters
                                     # are used in passwords.
                                     # Default is ENGLISH

HIDE_METHOD = NO_CHANGE | PRI | PRILOG | SELECT | PRI_TRUE | PRILOG_TRUE | SELECT_TRUE
                                     # Specifies the method for hiding partitions.
                                     # Starting with version 1.7.0 it is possible to
                                     # specify an individual hiding of partitions
                                     # (keywords SELECT or SELECT_TRUE)
                                     # Default is NO_CHANGE.
                                     #
                                     # PRI or PRI_TRUE will hide primary partitions
                                     # automatically. Logical partitions remain unchanged.
```



```
                                # PRILOG and PRILOG_TRUE will hide primary and
                                # logical partitions automatically.
                                #
                                # Remark:
                                # The previous values AUTO (same as PRI) and
                                # AUTO_TRUE (same as PRI_TRUE) are still recognized.

RESERVED_LBA = <nn>             # LBA sector in track 0 used by third-party program.
                                # When the boot manager is installed in the MBR (track 0)
                                # this sector will be skipped, thus the boot manager
                                # and the third-part program can both exist in the track 0.
                                # Default is 0 meaning that no sector will be skipped.
```

4.7.1.3 Keywords in section [ENTRY_<nn>]

```
NAME = <name>                   # Specifies the partition name (12 ch.).
                                # Default is empty string.

DESCRIPTION = <descr>           # Specifies the partition description (33 ch.).
                                # Default is empty string.

PASSWORD = <xxx>                # Password for partition (encrypted or plain text).
                                # Accepted only in licensed version.
                                # Default is empty string.

DRIVE = <xxx>                   # Specifies the drive where the partition resides.
                                # Allowed values are (1...N) or A.
                                # There is no default, the keyword is mandatory.
                                # The value A means booting from diskette A:.
                                # The values 1...N mean disk 1...N.

                                # The following keywords offer five different
                                # possibilities for specifying the partitions
                                # to be included in the boot manager.
                                #
LABEL = <xxx>                   # - The keyword LABEL identifies the partition
                                #   by the volume label.
PARTNR = <nn>                   # - The keyword PARTNR identifies the partition
                                #   by the partition number (see report).
PTABLE = <nn>                   # - The keyword PTABLE identifies the partition
                                #   by its partition table entry (0..3)
                                #   Usage if the keyword PTABLE is not recommended.
LBA = <nn>                      # - The keyword LBA identifies the partition
                                #   by its LBA start sector (0..N).
POSITION = <nn> MB | GB         # - The keyword POSITION identifies the partition
                                #   by its disk position. The specified position
                                #   must lie somewhere inside the partition.
                                # Remark: One of the keywords LABEL, PARTNR,
                                #         PTABLE, LBA or POSITION is mandatory for
                                #         identifying the desired partition
                                #         when booting from a disk.
                                # Remark: When booting from diskette A: the five
                                #         keywords LABEL, PARTNR, PTABLE, LBA and
                                #         POSITION are ignored.
                                # Remark: 1 MB = 1024 * 1024 Byte
                                #         1 GB = 1024 * 1024 * 1024 Byte

#
# The following keywords are used for hiding individual partitions.
# They are only allowed when HIDE_METHOD is set to SELECT or SELECT_TRUE.
```



```
#
# The keywords for specifying the partition to hide are similar to
# the keywords specifying the partitions to be booted.
# Only the prefix HIDEPART_<n>_ is added (see below),
# where <n> is a number between 1 and N (see below).
#
# Example:
# In order to hide three partitions selectively you would
# specify the prefixes HIDEPART_1_, HIDEPART_2_ and HIDEPART_3_.
#

HIDEPART_<n>_DRIVE = <xxx>          # The drive of the partition to be hid.

HIDEPART_<n>_LABEL = <xxx>          # One of these four keywords must be used
HIDEPART_<n>_PARTNR = <nn>          # in order to specify the partition to be hid.
HIDEPART_<n>_LBA = <nn>
HIDEPART_<n>_POSITION = <nn> MB|GB
```

4.7.2 Uninstall boot manager

The command for uninstalling the boot manager is:

```
bootusc bootmanager remove [batch]

Example:
bootusc bootmanager remove
```

With this command the boot manager is removed from the boot disk 1. The MBR code is replaced by the previous MBR code. In case the boot manager is installed in a primary partition this partition is deleted. The former active partition is re–activated. If necessary the first partition is set active. All hidden partitions on all disks are made visible. When `batch` is specified no user intervention is necessary.

4.7.3 Show status about installed boot manager

The command for displaying status information about an installed boot manager is:

```
bootusc bootmanager status

Example:
bootusc bootmanager status
```

When the boot manager is not installed on the boot disk 1 the error code 4 is returned.

The status function does not require a license.

4.7.4 Update entry in boot manager

The command for updating the name and/or description of a certain entry (see status command) in an installed boot manager is:

```
bootusc bootmanager update entry=n [name=<name>] [descr=<descr>] [partid=<id>] [batch]

  n  = 1..N or 'selected' (N = total number of entries, see status)
```



```
  id = XXh  (XX = two digit hex number)

Examples:
bootusc bootmanager update entry=1 name="Win XP" descr="New installation"
bootusc bootmanager update entry=1 partid=07h
bootusc bootmanager update entry=selected partid=07h
```

By this command the name <name>, the description <descr> and/or the partition ID <id> of the entry n in an already installed boot manager can be changed. All other data of the corresponding boot manager entry remain unchanged. Alternatively to a specific entry the currently selected entry can be changed. When batch is specified no user intervention is necessary.



4.7.5 Preselect other entry in boot manager

The command for changing the preselected entry (see status command) in an installed boot manager is:

```
bootusc bootmanager preselect entry=n [boot_once] [batch]

  n  = 1..N (N = total number of entries, see status)

Examples:
bootusc bootmanager preselect entry=2
bootusc bootmanager preselect entry=3 boot_once
```

With this command the preselected entry in an already installed boot manager can be changed to the specified entry. When `boot_once` is specified the specified entry is booted just one time, while the previously preselected entry is preserved. When `batch` is specified no user intervention is necessary.

The preselect function does not require a license.

The function for a one−time change of the preselected entry is useful especially when remotely accessing the machine. A typical usage pattern for maintenance purposes is to start a certain system just once and afterwards return to the standard system. This one−time change of the booted system can be achieved now easily by the parameter `boot_once`.



4.8 Repair MBR

In order to remove a boot manager which is installed in the MBR, a standard boot loader must be written to the MBR. The command for replacing the boot loader in the MBR is:

```
bootusc repair_mbr [file=<filename>]

Examples:
bootusc repair_mbr
bootusc repair_mbr file=track0.bin
```

The partition table remains essentially unchanged. The only modification of the partition table is that the first primary partition is activated if none is active.

When no file name is specified a standard boot loader is written in the MBR. This standard boot loader searches the partition table for an active partition and starts the boot sector of that partition. When a file is specified the boot loader is extracted from this file and written to the MBR.



4.9 Report about partitions

A report about all partitions on all disks can be displayed by the following command:

```
bootusc report [file=<filename>]

Example:
bootusc report
bootusc report file=info.txt
```

By default the report is displayed in the console window. When you specify a file name the report is saved in this file. The partitions are sorted according to their position on the disk. Primary and extended partitions have a partition numbered from 1 to 4. Logical partitions have a partition number of 5 and higher.

The report function does **not require a license**. This function is by intention not protected by a license.

The following screenshot (fig. 6) shows an example for the output of the report function:

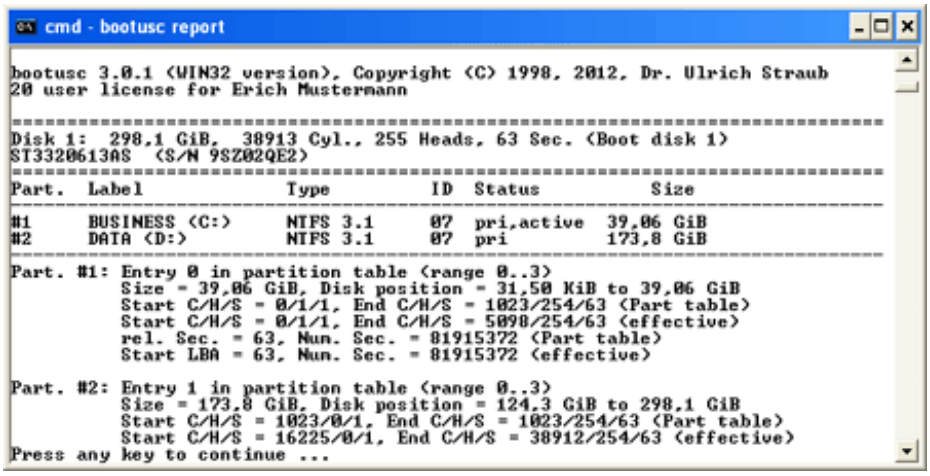


Fig. 6: Report about partitions



4.10 Encrypt passwords

The following command allows you to look up the encrypted form of a password:

```
bootusc encrypt_pw <passwd>

Examples:
bootusc encrypt_pw test
bootusc encrypt_pw "test 123"
```

The string <passwd> contains the plain text password. Valid characters in the plain text password are almost all printable characters of the 7 bit ASCII character set. The command displays the password in encrypted form. The opposite is not supported, i.e. there is no command provided which allows you to look up the plain text password from an encrypted password.

This command is available only with a license.



4.11 Reboot the computer

The following command allows you to reboot the PC:

```
bootusc reboot [batch]

Example:
bootusc reboot
```

When `batch` is specified the user is not prompted for confirmation.



4.12 Exit codes of command–line program

The command–line program `bootusc.exe` returns the following exit codes:

```
0 = Command executed successfully
1 = Syntax error or invalid parameters specified (command was not executed)
2 = Command aborted by user
3 = Error during execution of command
4 = Boot manager is not installed (only for command 'bootmanager status')
```



5 Tips

Please read the following tips **before** you install the boot manager to hard disk or diskette! These recommendations will help you to avoid potential trouble when setting up a multi−boot system.

General:

1. **In which cases one might face problems ?**
2. **Recommendations for safe usage of Boot−US**
3. **Uninstalling the boot manager Boot−US**

The remaining tips contains answers to frequently asked questions. Especially the Windows tips 1 and 2 explain how multiple versions of Windows may be installed and booted by the boot manager of Boot−US.

Miscellaneous:

1. **Basic questions about Boot−US**
2. **Unhiding partitions in the boot manager**
3. **Error messages on integrity checks**
4. **Double occupied sectors in track 0**

Installation:

1. **Limitations of different operating systems**
2. **Check independence and completeness**

Windows:

1. **Installing Windows several times on one disk**
2. **Installing Windows several times on different disks**
3. **Boot−US does not detect SCSI or IDE disks under NT**
4. **Support for Windows 2000/XP/2003/Vista/2008/7/8**
5. **True partition hiding for Windows 2000/XP/2003/Vista/2008/7/8**
6. **Copying (cloning) a Windows 2000/XP/2003 installation**
7. **Administrator rights under Vista and Windows 7**
8. **Windows 7**
9. **Windows 8**

Linux:

1. **Booting Linux by the boot manager of Boot−US**
2. **Linux partition is recognized by Boot−US as non−bootable**



5.1 In which cases one might face problems ?

Disk managers:
When disk managers are installed on the first hard disk e.g. the OnTrack or EZ−Drive disk manager, then an installation of the boot manager Boot−US to the MBR may cause **problems**. Typically disk managers are used in older PCs which are without LBA support for accessing disks larger than 500 MiB. When the mainboard BIOS supports EIDE disks or offers LBA access disk managers are usually neither required nor installed.

The **incompatibility** of a boot manager (when installed on the **MBR**) with disk managers has two reasons. Firstly, both programs are installed potentially on the same sectors, i.e. on the MBR and the beginning of the track 0. Secondly, when the boot manager Boot−US is installed on the MBR it becomes active before the disk manager. This means the boot manager is not supported by the disk manager and therefore the boot manager can access the wrong sectors of the disk.

When the boot manager is installed on a **primary partition** the above mentioned problems should not arise. In this case the MBR contains the necessary code to start the disk manager before the boot manager becomes active, and both programs reside on different sectors.

ABIT RAID adapters:
The following warning does not apply to the case where the RAID adapter is used just as a fast IDE channel and no true RAID functions are active. In this case there are no RAID configuration data which would have to be stored.

If RAID functions are used the corresponding RAID configuration data must not get destroyed, otherwise, you won't be able to access the data on the RAID disks any more. It seems that the ABIT UDMA RAID adapter stores its configuration data directly on user accessible sectors on the hard disk, i.e. on the track 0. This way of storing the configuration data is extremely unsafe. The data can be easily overwritten, for example, when the boot manager Boot−US is installed on the MBR, or when a disk manager is installed on the MBR. All restrictions and warnings about disk managers also apply to the ABIT RAID adapater. For example, you must **not install** the boot manager of Boot−US to the MBR when using an ABIT RAID adapater.

When the boot manager is installed on a **primary partition** problems should not arise.

Complicated configuration:
The following cases are classified as complicated configuration:

- Disk number of boot partition has been changed
- Partitions are booted over several steps
- Partitions are not independent

In such cases you should be careful when you install the boot manager Boot−US. The boot process is quite different from the one for which Boot−US has been designed. Boot−US assumes that it is sufficient to simply load the boot sector of the desired partition from disk and start it **unchanged**.

When the disk number of the boot partition is **changed**, for example if a new disk is inserted or removed, it is necessary to specifically modify the boot sector in order that it can still be used to boot the partition. For example, this happens when Windows 95/98/ME is installed on the first disk and another disk is installed ahead of the first disk. The former first disk now becomes the second disk and Windows 95/98/ME now resides on the second disk. Boot−US does contain special support for booting **DOS** and **Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8** from a "second" hard disk. For other operating systems a similar support might be necessary but it is not yet present in Boot−US.



The other problematic case is when **more than one** partition is involved in the boot process. When the boot manager is installed care must be taken that all partitions involved **remain visible** during the boot process. The boot manager of Boot–US has no problems at all in loading the boot sector of a hidden partition but the respective operating system probably will **fail** to start when the boot partition is **hidden**.



5.2 Recommendations for safe usage of Boot–US

Please pay attention to the following recommendations. They will help you to avoid problems when setting up a multi–boot system. If you face a problem please try to find the cause of the problem and then take the appropriate measures. Even if at first glance the whole system seems to be corrupt usually only a **small** part of the data is changed. In many cases everything can be repaired easily with standard tools. A complete re–installation of the system is definitely the very last option which is appropriate only in extreme cases.

Backup your data
The most important rule is to **backup all important data** on independent storage media. If something goes wrong during the installation of another operating system or the boot manager and a new installation of the operating system is necessary then you can restore your data, although this would require a considerable amount of time.

Default settings
The default settings in the dialogs of Boot–US up to version 1.1.4 were aimed at maximum convenience for an experienced user. Beginning with version 1.2.1 the default settings were changed to offer maximum safety also for an unexperienced user. Since the default settings in different version are different, you cannot blindly rely on these default settings. Please check them according to the following recommendations.

Any questions ?
If you have any question or something is not clear to you, please try to resolve your queries before you continue. The chapter **glossary** explains the important terms which you will come across when setting up a multi–boot system.

Is the information displayed correctly ?
When Boot–US starts up it collects the information about all partitions on all hard disks and displays that on the initial screen. Please check whether all partitions are displayed correctly and whether the disk parameters (number of cylinders, heads, sectors and capacity) have been correctly determined. If there is something wrong or if there are any error messages issued by Boot–US **do not install** other operating systems or the boot manager.

Save partition and boot sectors
Before you begin to experiment with the boot manager it is strongly recommended that you save the list of all partition and boot sectors from all disks to a bootable diskette. The created file contains **all** sectors which might get modified by the boot manager installation and operation. These sectors can be saved and restored both by the Windows program Boot–US and the DOS command–line utility **bootusc.exe**. It is recommended to save these sectors with both programs to two different files. The contents of both files must be the **same**.

Installation to diskette
It is strongly recommended to install the boot manager first to diskette and not on the hard disk. It is important that you choose the setting **"Do not change partitions IDs"**. This is the default setting for version 1.2.1 and later, but not in earlier versions! With this settings no partition IDs are changed. After the boot manager has been installed to diskette you can verify that no data has been changed on the hard disk. Just restart your system and you will notice that everything works as before. Afterwards you can boot the configured partitions by the boot manager residing on the test diskette. As long as no partitions are hidden you can boot all these partitions by the test diskette.

Installation in primary partition
When the installation to diskette is successful one can think of installing the boot manager Boot–US to a **primary partition** on the first hard disk. Please note that this target is the default setting only for version 1.2.1 and later. Thus you need to **perhaps change** the installation target. It is also strongly recommended to save the complete track 0 of the first hard disk to a file and copy that file to a standard DOS formated diskette. Part of this file is the MBR with the partition table of the first hard disk. This allows you to restore the MBR of the first hard disk with a disk editor or with



the DOS program **bootusc.exe**. Please again use the setting **"Do not change partition IDs"** when you install the boot manager Boot–US to a primary partition. As noted above this is the default setting for version 1.2.1 and later. If something goes wrong when installing the boot manager in a primary partition and Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 cannot be booted any more, you can restore the previous state by standard tools. By a DOS boot diskette and a standard partition manager you can deactivate the boot manager partition and activate the partition which was active before.

Installation in MBR
You should install the boot manager of Boot–US to the MBR only if this is necessary i.e. when you need the boot manager partition for other purposes. The advantage of the installation in the MBR is that no additional partition is occupied. The disadvantage is that for repairing problems more effort is required. If a problem appears and Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 cannot be booted anymore the boot manager of Boot–US must be removed from the MBR with the standard tool FDISK or by the DOS program bootusc.exe, see the following tip Removing the boot manager in case of problems.



5.3 Uninstalling the boot manager Boot−US

The boot manager Boot−US can be uninstalled by any of the following three methods:

- command **Bootmanager / Remove...** in the configuration program Boot−US (GUI)
- command **bootusc bootmanager remove** in the command−line version of the configuration program Boot−US
- command **Uninstall boot manager** directy in the boot manager

All three methods are equivalent. The same operations are performed in all three cases. In short, the boot loader in the MBR is restored, a potentially created partition for the boot manager is deleted and the former active partition is reactivated. Additionally all hidden partitions are made visible.

Uninstalling a boot manager manually

Sometimes it might be necessary to uninstall a boot manager by standard tools, especially when the accompanying uninstallation program is not available. As explained you should never face such a situation with the boot manager of Boot−US. However with respect to other boot managers it might be useful to know how to remove them by standard tools. The necessary steps depend mainly on the location where the boot manager is installed.

Boot manager in separate primary partition
This type of installation is very easy to remove with standard tools. It is sufficient to boot from a DOS diskette and deactivate the boot manager partition with any partition manaer and activate another partition. Examples for this type of installation are the OS/2 boot manager or the boot manager Boot−US when installed on a separate partition.

Boot manager in MBR
This type of installation can also be removed with standard tools. Again a DOS boot diskette with the standard FDISK program is needed. By the command **fdisk /mbr** a standard boot loader code is written to the MBR. The purpose of the standard boot loader code is to search the partition table for a partition marked as active and to boot that partition. After that by a partition manager an appropriate partition has to be marked as active. Alternatively by issuing the command **bootusc repair_mbr** an equivalent standard boot loader code can be written to the MBR. In the latter case the first partition is activated automatically. An example for this type of installation is the boot manager of Boot−US when installed on the MBR.

Boot manager in operating system
In this case the boot manager is integrated into the operating system. Naturally there is no general rule for removing such a boot manager. It might even be impossible to remove such a boot manager. The boot manager becomes active only after the operating system has been started. It must be removed by the means offered by the corresponding operating system. An example for this type of installation is the file "boot.ini" of Windows NT/2000/XP/2003. This boot manager cannot be removed since it is necessary for starting Windows NT/2000/XP/2003.

Hidden partitions
When the boot manager is removed by the configuration program Boot−US then all hidden partitions will be made visible during the uninstallation process. However if the the boot manager is removed "manually" hidden partitions will remain hidden. Usually the hex number 10h is added to the partition ID in order to hide a partition. This way of hiding a partition is used, for example, by Boot−US and the OS/2 boot manager. In these cases the partitions can be made visible simply by issuing the command **bootusc unhide**. Some boot managers also change the boot sector when hiding partitions. Such partitions cannot be made visible by standard tools. The original program used for hiding the partition must be employed for unhiding them.



During the test of a boot manager it might happen that partitions are hid –– –– intentionally or unintentionally. Meanwhile this can be easily repaired directly in the boot manager, see Miscellaneous tip 2.



5.4 Basic questions about Boot−US

Does Boot−US work under a 64 bit Windows?
The answer is yes. From a technical point of view all Windows components of Boot−US are 32 bit components. Clearly this means that these components will work under a 32 bit Windows. However all 64 bit Windows systems allow 32 bit Windows programs to execute, hence Boot−US can also execute under a 64 bit Windows system. Boot−US does not work under the very old 16 bit Windows like Windows 3.x.

Explanation of version numbers
The first two digits of the version number of Boot−US represents the version of the accompanying boot manager. The third digit counts the fix levels of a certain version. Each time when the data structure for storing the configured partitions inside the boot manager is changed, a new version number is assigned to the boot manager. For example, the version number 1.2.2 of Boot−US tells you that three versions −− 1.0, 1.1, 1.2 −− of the boot manager were created. A higher version of Boot−US always understands (reads) all previous versions of the boot manager. When you upgrade Boot−US it is therefore **not** required to uninstall the boot manager before.

What is a trace file?
A trace file contains a protocol of the important function calls. The trace output is built by the developer into the program. The developer has specified the important function calls to be traced. The trace file is mainly a means for problem determination. In case of problems one can see which function calls were executed on the client machine. From the trace file the problem is either identified directly or at least one can obtain hints about the cause of the problem. Inside Boot−US the trace output can be activated by the menu command **Configuration / Settings...**.

How do I install the license?
The license for Boot−US is delivered in form of a license file **`bootus.lic`**. This license file must be copied (unchanged!) into the installation directory of Boot−US. The file name must be `bootus.lic`. When Boot−US finds a valid license file at start−up in the installation directory all features of Boot−US are enabled.

Inside Boot−US it can be easily verified whether a valid license is installed. Firstly, the window's title changes from "Boot−US [non−licensed version]" to "Boot−US". Secondly, the menu command **Help / About Boot−US...** directly shows the current number of licenses and the licensee.



5.5 Unhiding partitions in the boot manager

When the boot manager is tested and some partitions could not be booted any more, the typcial reason is that during this test, some partitions have been **hidden** either intentionally or unintentionally.

Unhide partitions in the boot manager
Starting with version 1.2.1 of Boot−US these hidden partitions can be made visible **very easily** directly in the boot manager Boot−US. Please press the **F10** key in the boot manager. This opens a configuration menu where you can execute the command **Unhide partitions on all disks**. This makes all hidden partitions on all disks visible and allows access to all partitions as before.

Comment:
Starting with Boot−US 1.5.1 this command also makes **true hidden** partitions visible.



5.6 Error messages on integrity checks

When reading the list of all partitions from all disks an automatic integrity check of all partitions is carried out. As a result of these checks the following warning or error messages could be displayed for individual partitions. Warnings are marked as W<nnn> and error messages as E<nnn>:

E001: MBR of disk <xx> is invalid: signature 0xAA55 missing.
Either the disk is empty or the MBR is corrupt.

E002: # heads and # sec. of disk <xx> do not agree with partition table.
When the partition was created different disk parameters were set. You must reset the disk parameter to the previous values.

E003: partition sector (LBA=<xx>) invalid: signature 0xAA55 missing.
Either the partition sector is corrupt or the address of the partition sector is wrong.

E004: boot sector (LBA=<xx>) invalid: signature 0xAA55 missing.
Either the boot sector is corrupt or the address of the boot sector is wrong.

E005: MBR of disk <xx> contains more than one extended partition.
The extended partition was created with a partition manager which violates the established standards. You should use a partition manager which sticks to the established standards.

E006: log. partition sector (LBA=<xx>) invalid: signature 0xAA55 missing.
Either the logical partition sector is corrupt or the address of the logical partition sector is wrong.

E007: log. partition sector (LBA=<xx>) invalid: found two partitions in partition sector.
The logical partition was created with a partition manager which violates the established standards. You should use a partition manager which sticks to the established standards.

E008: log. partition sector (LBA=<xx>) invalid: found two extended partitions in partition sector.
The logical partition was created with a partition manager which violates the established standards. You should use a partition manager which sticks to the established standards.

E009: log. partition not fully contained in extended partition.
The logical partition was created with a partition manager which violates the established standards. You should use a partition manager which sticks to the established standards.

E010: boot sector is true hidden, but ID=<xx> represents unhidden partition.
The corresponding partition was unhid incompletely. This error can be solved easily by hiding or unhiding the partition a second time.

E011: partitions overlap. There is a serious danger to lose data!!
Sooner or later this error will lead to data corruption. You must delete one of the two partitions in order to prevent the partitions from overwriting each others data. The respective partitions were created by an erroneous partition manager. You should use a partition manager which is working accurately.

E012: LBA values from partition table and effective values do not agree.
The partition was created with a partition manager which violates the established standards. You should use a partition manager which sticks to the established standards.



W001: NTFS format of partition is invalid.
The NTFS format of the partition seems to be corrupt.

W002: CHS values from partition table and effective values do not agree.
This warning indicates a non−severe problem which can be ignored for the moment.

W003: partition does not start and end on cylinder boundaries.
This warning indicates a non−severe problem which can be ignored for the moment.

W004: partition does not start on cylinder boundaries.
This warning indicates a non−severe problem which can be ignored for the moment.

W005: partition does not end on cylinder boundaries.
This warning indicates a non−severe problem which can be ignored for the moment.

W006: partition is not aligned on 1 MiB boundary.
This warning indicates a non−severe problem which can be ignored for the moment.



5.7 Double occupied sectors in track 0

It can happen that third–party programs store data in indiviual sectors of track 0. Especially some Adobe programs seem to store copy protection data in sectors of track 0. If the boot manager is installed in the MBR and the following sectors of track 0 it can easily happen in this case that a sector is occupied by the third–party program as well as by the boot manager.

It is clear that such a **double occupany** will lead to **problems**, especially the boot manager will not start correctly any more.

Solution:

When installing the boot manager into the MBR it is now possible to specify a sector which is occupied by a third–party program. This sector will then not be used by the boot manager.

The sector reserved for the third–party program is specified by the following parameter `RESERVED_LBA` in the file `bmgrus.ini` (see chapter Contents of bmgrus.ini):

```
[GLOBAL]
RESERVED_LBA = <nn>
```

<nn> is the LBA sector number of the reserved sector.

Which sector has been modified?

The position of the double occupied sector can be determined as follows:

- save track 0 immediately after boot manager has been installed in MBR
- save track 0 after the third–party program has written data to the track 0 (and corrupted the boot manager)
- compare the two backup files to see which sector has been changed



5.8 Limitations of different operating systems

Preliminary remark: The following limits apply to a disk with 255 heads and 63 sectors per track. If the actual disk parameters are smaller, the limits might need to be adapted appropriately. It is therefore recommended to configure the disk with these disk parameters (255 heads and 63 sectors per track). A later modification of the disk parameters is **not** possible without reinstalling all partitions!

DOS with FAT16:
The beginning of a bootable DOS partition must lie within the first 2 GB of a disk, in order to successfully boot that partition.

Windows 95/98 with FAT16/FAT32:
The beginning of a bootable Windows 95/98 partition must lie within the first 8 GB of a disk, in order to successfully boot that partition.

Windows ME with FAT16/FAT32:
The 8 GB limit does not exist any more for Windows ME. A bootable Windows ME partition can be created anywhere on the disk. If the partitions begins beyond the 8 GB limit, the BIOS must support the extended INT 13h in order to boot the partition successfully.

Windows NT 4 with FAT16:
The beginning of a bootable Windows NT 4.0 partition with the file system FAT16 must lie within the first 2 GB of a disk, in order to successfully boot that partition.

Windows NT 4 with NTFS:
The beginning of a bootable Windows NT 4.0 partition with the file system NTFS must lie within the first 4 GB of a disk, in order to successfully boot that partition.

Windows 2000/XP/2003/Vista/2008/7/8:
The 8 GB limit does not exist any more for Windows 2000/XP/2003/Vista/2008/7/8. A bootable Windows 2000/XP/2003/Vista/2008/7/8 partition can be created anywhere on the disk. If the partitions begins beyond the 8 GB limit, the BIOS must support the extended INT 13h in order to boot the partition successfully.

Linux:
The 8 GB limit does not exist any more for Linux starting with LILO 21.6. A bootable Linux partition can be created anywhere on the disk. If the partitions begins beyond the 8 GB limit, the BIOS must support the extended INT 13h in order to boot the partition successfully.



5.9 Check independence and completeness

The boot manager Boot−US allows you to boot and use multiple operating systems independently of each other. This requires, however, that the operating systems have been installed independently and completely each into a separate partition. When a new operating system is installed (see chapter Tips / section Windows) this is automatically the case.

Frequently certain operating systems are already installed. Especially with Microsoft operating systems the standard installation of a second, third,... operating system does **not** result in an independent and complete installation into a single partition.

Test:

You must check whether an operating system installed in a primary partition can be booted independently from the remaining operating systems respective partitions (without using an external boot manager). For this check please follow the instructions below:

- The partition to be checked has to reside on the first boot disk. If that is not the case, please interchange the order of the boot disks in the BIOS. Alternatively you could temporarily detach the disks residing before the one to be checked.
- The primary partition to be checked has to be set active.
- All other primary and logical partitions on the same disk must be true hid.
- Restart the PC.
- The partition to be check should start up successfully without using any external boot manager.

Recommendation: All partitions to be booted by the boot manager Boot−US should be checked whether they pass the above test. Partitions with operating systems which pass this test are installed completely and independently. They can be booted without problems by the boot manager Boot−US. Operating systems which do not pass this test and are to be booted by Boot−US, have to be repaired first.



5.10 Installing Windows multiple times on one disk

Overview:
DOS, Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 can be installed multiple times without any problems onto different primary partitions on the first hard disk. In principle you have the choice between two setup strategies:

- Don't hide already existing partitions
- Hide the already existing partitions before installing a new one

The first strategy is the method used by **Microsoft**. In this case there is a central partition which is used for booting the other partitions. It is obvious that all partitions are implicitly connected by this central partition. When this central partition is deleted or changed **all** other primary partitions are affected, i.e. they might become unbootable. Furthermore the central partition must always reside on the first hard disk.

The second strategy is the method **recommended** by the author of Boot−US. This type of installation requires a litte more effort and planning, but it has the **advantage** that all installed partitions are truly independent of each other. This independence results simply from the fact that all already existing partitions are hidden when the next partition is installed. Therefore this new installation appears exactly like the first installation. Due to the independence any partition can later be modified or deleted without affecting the other partitions.

Remark: Please make sure that you know the installation limits of all involved operating systems.

Installation example:
Windows XP is to be installed twice in two separate primary partitions on the first hard disk. Both partitions should be booted by the boot manager of Boot−US. This can be achieved as follows:

- Install Windows XP on a primary partition.
- For safety reasons install the boot manager Boot−US to diskette or CD with activated "automatic true hiding of partitions". With this diskette or CD this first Windows partition can be booted even when it is true hidden.
- True hide the Windows partition by means of Boot−US. The menu command **Partitions / Details...** opens a dialog where single partitions can be true hid.
- Create a second primary partition and set it active. This can be achieved by the configuration program Boot−US (GUI) or any other partition manager.
- Install the second copy of Windows in this second primary partition. The first Windows should not be running when the second installation is started. The first Windows has been true hidden before and therefore it cannot disturb the second installation.
- Start the configuration program Boot−US (on the second Windows) and install the boot manager with activated "automatic true hiding of partitions". Both Windows partitions can be included in the boot manager.
- Restart the PC and select the desired partition.
- The selected partition will be made visible by the boot manager and the other one will be true hid.

By the same procedure other combinations of DOS, Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 can be installed multiple times on the first disk.

Comment:
If Windows 2000 or higher is involved you must true hide all partitions, see Windows tip 5. If only DOS or Windows 95/98/ME or NT is involved then simple hiding of partitions is sufficient.



5.11 Installing Windows multiple times on different disks

Remark 1: Please make sure that you know the installation limits of all involved operating systems.
Remark 2: The disk parameters "number of heads" and "sectors per track" must not change between installation of an operating system and the later usage. The simplest solution is to configure in the BIOS all disks to have 255 heads and 63 sectors per track.

In order to install DOS and Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 multiple times on different disks the following procedure should be employed:

- DOS, Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 is installed on the first disk.
- A physical second disk must appear temporarily as the first disk. This can be achieved, for example, by deactivating temporarily the physical first disk (detach the data cable).
- On this physical second disk (now appearing as first disk) DOS or Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8 can be installed without problems.
- The physical first disk is re−activated. Both disks are now available as before.
- Windows on the first disk is booted (without support of Boot−US).
- The configuration program Boot−US is started and the desired partitions on the first and second disk are included in the boot manager. Again "automatic hiding of partitions" must be activated.
- By the boot manager Boot−US all partitions on the first and second disk can be booted. The selected partition will be made visible while the other non−selected primary partitions will be hid during the boot process.

Comment:
If Windows 2000/XP/2003/Vista/2008/7/8 is involved you must true hide all partitions, see Windows tip 5.



5.12 Boot–US does not find SCSI or IDE disks under NT

Under Windows NT on a system with **mixed** disk types (EIDE and SCSI) it could happen that Boot–US detects only one disk type (SCSI or EIDE). This is not a failure of Boot–US, but it is caused by Windows NT. You could verify that NT has the same problem by starting the Disk Administrator of Windows NT. This program will also **not detect** the same disk type. The problem of the "apparent missing" disks is caused by non–installed or non–started drivers for the corresponding disk type. The following hints are intended to help you solving this problem.

Install (E)IDE drivers:
The driver `atdisk.sys` handles IDE disks while the driver `atapi.sys` handles EIDE disks. The instructions to install or start the driver `atdisk.sys` for IDE disks (see also Microsoft Knowledge Base article Q133491) are as follows:

- The driver `atdisk.sys` must reside in the directory `%systemroot%\system32\drivers`.
- In Control Panel / Devices set the start mode of the device ATDISK to "System". If you want that the newly added drives appear first in the Disk Administrator set the start mode to "Boot".
  Attention: changing the start mode could prevent NT from booting.
- Start the device ATDISK.
- Start the Disk Administrator of NT and assign appropriate drive letters to the partitions of the (E)IDE and SCSI disks. If you do not assign drive letters, Windows will choose them automatically at the next startup and they might be different from the current drive letters.
- Restart Windows NT.

Install SCSI (and EIDE) drivers:
SCSI drivers are provided usually with the SCSI controller. The installation is straightforward:

- Open Control Panel / SCSI Adapters and click on Drivers / Add....
- A dialog box opens. Select the appropriate model of the SCSI controller.
- Insert the installation CD or diskette in the approriate drive.
- NT copies the necessary driver files from the CD or diskette.

Remark: (E)IDE drivers can also be installed with this approach.



5.13 Support for Windows 2000/XP/2003/Vista/2008/7/8

Starting with **version 1.2.4** Boot−US supports **Windows 2000/XP/2003/Vista/2008/7/8**. In order to install under Windows 2000/XP/2003/Vista/2008/7/8 the boot manager Boot−US you **must** use the version Boot−US 1.2.4 or higher. If you install under Windows 2000/XP/2003/Vista/2008/7/8 the boot manager using Boot−US 1.2.3 (or earlier) you may suffer serious **data loss**!

Techn. background:
Microsoft has changed in Windows 2000/XP/2003/Vista/2008/7/8 the WIN32 function for accessing the hard disk. If the disk is accessed by the old functions (of Boot−US 1.2.3), the access should be denied always and an error should be returned. Unfortunately this error checking does not work reliably in Windows 2000/XP/2003/Vista/2008/7/8. Under certain circumstances Windows 2000/XP/2003/Vista/2008/7/8 does **allow** the disk access with the "old" functions calls. Write access with the old function calls seems to work, i.e. no error is returned. In reality, however, **incorrect** data is written to the disk resulting in the mentioned data loss.



5.14 True partition hiding for Windows 2000/XP/2003/Vista/2008/7/8

Windows 2000/XP/2003/Vista/2008/7/8 ignores the simple hiding of partitions, i.e. those partitions are still visible to Windows 2000/XP/2003/Vista/2008/7/8. In order to hide a partition under Windows 2000/XP/2003/Vista/2008/7/8 the partition must be **true hid**. When a partition is true hid the partition ID and the boot sector of the respective partition is modified. True hiding a partition is available with Boot−US 1.5.0 (or higher).

Comment:
If Windows 2000/XP/2003/Vista/2008/7/8 is involved and multiple independent installations of Windows are to be performed, the present partitions must be **true** hid before the next installation is carried out. The true hidden partitions are ignored by all versions of Windows (incl. Windows 2000/XP/2003/Vista/2008/7/8).

Comment:
It is possible that Windows 2000/XP/2003/Vista/2008/7/8 considers a true hidden partition as **unformatted**. This can be changed by **removing the drive letter** from a true hidden partition. This prevents that you accidentally format such an "unformatted" partition thereby deleting all existing data on that partition. Drive letters are assigned or removed within the Windows disk administrator (Computer Management / Storage / Disk Management).



5.15 Copying (cloning) a Windows 2000/XP/2003 installation

Copying or cloning a partition with Windows 2000/XP/2003 does not always lead to second independent Windows installation. Especially when the image is installed on the same hard disk, the second Windows becomes dependent on the original partition. On the other hand, copying the image to another disk seems to work better.

The reason for these problems is that all data are copied unchanged. Especially the absolute partition identifiers are also duplicated. This means the image contains references to the original partition. When the image is booted, Windows follows the references and tries to access the original partition and not the image. If the image resides on another disk, it seems that Windows considers the references as invalid and accesses the image correctly.

These problems can be avoided by deleting the following registry key before creating the image:

`HKEY_LOCAL_MACHINE\System\MountedDevices`.

Using this measure the references to the original partition are missing in the image. When the original partition or the image is started the next time Windows recognizes that this registy key is missing and reconstructs it from the existing partitions.

This tip has been kindly provided by Christian Wittmer from **Scorpio IT** (**http://www.scorpio–it.com**). Thank you very much.

Attention:
When trying the above tip it is strongly recommended to backup the registry before deleting the registry key. When Windows has not been installed on C: severe problems have been reported after deleting the registry key. In general you are using this tip at your own risk.



5.16 Administrator rights under Vista and Windows 7

Starting with Vista Microsoft has changed the user rights management. These changes also apply to Windows 7. Even if a user is logged on as administrator programs started by this user are running without administrator rights by default.

Microsoft has planned that programs are running initially without administrator rights even if they are started by an administrator. When the program requires higher rights for a certain operation these rights are requested interactively. This allows the user to recognize and control which operations are executed with administrator rights.

However it is more convenient to avoid the many confirmations for raising the user rights and start a program like Boot−US directly with administrator rights. This can be achieved as follows:

- Explorer: right click on program, then select "run as administrator" from the context menu
- Command−line window: press the key combination "<Ctrl> + <Shift> + <Enter>" to start the program



5.17 Windows 7

When you install Windows 7 on an empty disk then it is installed in two partitions: a 100 MiB boot partition and the actual Windows 7 partition. Windows 7 is booted by the 100 MiB partition. This partition must be inserted into the boot manager. Additionally you need to make sure that both partitions are simultaneously made visible or true hid. In other words: you need to use the individual true hiding of partitions, hence you must specify yourself which partitions need to be true hid in each case. The 100 MiB partition is used primarily for a BitLocker encryption. If you do not make use of BitLocker then it is easier to install Windows 7 in a single partition.

When Windows 7 is installed in two partitions then the automatic true hiding cannot be used, since in this case only one of the two Windows 7 partitions would be made visible while the other one would remain true hid. In such a case booting does fail since the Windows 7 installation is not completely accessible.

In order to install Windows 7 in a single partition you can create the target partition and set it active before starting the installation. The installation will then be carried out into the active partition. Other primary partitions on the same disk should, as usually, be true hid before starting the new installation.

Also without that preparation of the disk it is possible to install Windows 7 in a single partition, for example see the following link:
http://www.mydigitallife.info/hack–to–remove–100–mb–system–reserved–partition–when–installing–windows–7/comment–
The reference to this link was kindly provided by Jörg Staudte (www.musikcomputer.de).



5.18 Windows 8

Installation in BIOS mode:
If you want to setup a multi boot system with Windows 8 and Boot−US then you need to install Windows 8 in the BIOS mode. In this case the boot process of Windows 8 is similar to that of Windows 7. On an UEFI motherboard the BIOS mode is often called CSM (Compatibility Support Module) mode.

Installation in UEFI mode:
Boot−US does not support the UEFI mode. If you install Windows 8 in the UEFI mode then you cannot use Boot−US.



5.19 Booting Linux by the boot manager of Boot−US

Linux can be installed on a primary or logical partition on any hard disk. In order that such a partition can be booted by the boot manager of Boot−US the Linux loader LILO must be installed onto the **boot sector** of the Linux partition.

Below you will find a concrete example for the configuration file `/etc/lilo.conf` which has been kindly provided by a user (Sebastian Steinmetz) who boots Linux by the boot manager of Boot−US. In this case Linux is installed on a logical partition on the second EIDE disk (`boot=/dev/hdb5`). When Linux resides on another location the installation target of LILO must be modified appropriately. Starting with LILO 21.6 the Linux boot partition may reside beyond 8 GB limit (parameter `lba32`).

```
Example for /etc/lilo.conf:

# LILO configuration file

# Start LILO global Section
boot=/dev/hdb5           # installation target of LILO
compact                  # faster (not guaranteed to work)
read-only
prompt
timeout=1
vga = normal             # force sane state
lba32                    # overcome the 8 GB limit
# End LILO global section

image = /boot/vmlinuz    # image to boot
  root = /dev/hdb5       # root partition
  label = Linux          # appears at LILO prompt
  alias = l
```

After each change of this configuration file the modifications must be written back to disk by executing `/sbin/lilo`.



5.20 Linux partition is recognized by Boot−US as non−bootable

Boot−US considers a Linux partition as **non−bootable** when the Linux partition does **not** have a **valid boot sector**. Such a partition is indicated as non−bootable by Boot−US. Boot−US requires a valid boot sector in order to boot any partition. Contrary to Windows partitions there are several ways to boot a Linux partition. A missing boot sector is therefore nothing special for a Linux partition.

This problem of a missing boot sector can be easily repaired. It is only required to install the Linux loader LILO onto this partition (for example hdb5). In the previous Linux tip 1 you can find an example for the configuration file `/etc/lilo.conf`, where Linux and LILO are installed on the logical partition hdb5 on the second disk.





6 Glossary

The following glossary explains the most important expressions which you will always come across in regard to boot managers and partitions.

1. **Purpose of boot manager**
2. **Partition / partition sector / boot sector**
3. **Format of partition sector**
4. **Master Boot Record (MBR) / track 0**
5. **Primary / extended partition**
6. **Logical drive**
7. **Active partition**
8. **Hidden partition / true hidden partition**
9. **Bootable partition**
10. **Which partition IDs are recognized ?**
11. **LBA and CHS format, LBA mapping**
12. **INT 13h / extended INT 13h**
13. **SI and IEC units**



6.1 Purpose of boot manager

The purpose of a boot manager is to allow a user to boot different operating systems from the same PC (one at a time). Many operating systems contain an integrated boot manager, for example by the file "boot.ini" on Windows NT or LILO and GRUB for Linux. The main disadvantage of these built−in boot managers is that they can be used only after the corresponding operating system has been started. After that the actually desired operating system can be booted. Therefore, in many cases, two boot processes are necessary to start the target operating system. Also the configuration of these boot managers often requires detailed system knowledge.

On the contrary the boot manager Boot−US allows the selection of the configured operating systems directly after starting the machine before any operating system is booted. Hence the boot manager Boot−US is faster and more flexible than the built−in boot managers. Additionally it is very easy to configure the boot manager by the Windows configuration program Boot−US.



6.2 Partition / partition sector / boot sector

Partition:
A partition is a set of adjoined sectors on a hard disk. Any sector of the hard disk belongs either to no partition or it belongs to a specific partition, but it must not belong to more than one partition at the same time. In other words, partitions must not overlap each other. Typically, different operating systems are installed on separate partitions. An operating system can access all sectors of its partition, while access to sectors of other primary partitions is not possible in many cases. On the other hand, sectors of supported logical drives can be accessed without any problems.

There is the convention that partitions should always start and end on **cylinder boundaries**. No rule is without exception: since the first sector of a hard disk (C/H/S=0/0/1) is always reserved for the MBR and one does not want to leave the complete zeroth cylinder empty, the first partition does not begin on the next cylinder boundary (C/H/S=1/0/1) but directly after the track 0 at C/H/S=0/1/1. Hence only the sectors of the track 0 are left empty (with the exception of the first sector). There is no compelling technical reason for this convention. Linux and Windows NT/2000/XP/2003/Vista/2008/7/8 seem to have no problems when partitions do not start or end on cylinder boundaries. However other operating systems (e.g. DOS) expect that this convention is obeyed.

Partition sector:
Every partition requires an accompanying partition sector. The partition sector describes this partition, i.e. it contains the information about begin and end of the partition and the type of the partition. All partition sectors have the same structure, see format of partition sector.

The MBR is the common partition sector for primary and extended partitions. On the contrary, to every logical drive belongs a separate partition sector, which describes on one hand the logical partition and on the other hand contains the position of the partition sector for the next logical drive.

Boot sector:
The boot sector is the sector at the beginning of a bootable operating system partition. The boot sector contains a small program. When this program is loaded and executed the corresponding operating system is booted. The contents and the format of the boot sector depends on the operating system. A certain boot sector can therefore be used only for booting a particular operating system but not for booting any other operating systems.



6.3 Format of partition sector

A partition sector describes one or more partitions, i.e. it contains the information about the position, size and type of the partitions.

All partition sectors have exactly the same format (see table below). It should be noted, however, that the boot loader code is present only in the MBR partition sector. In the partition sectors for logical drives the boot loader code is empty.

```
------------------------------------
Partition sector (length 512 bytes)
------------------------------------
Offset  Bytes  Meaning

 000h    446    boot loader code
 1BEh    16     1. partition entry
 1CEh    16     2. partition entry
 1DEh    16     3. partition entry
 1EEh    16     4. partition entry
 1FEh    2      signature (55h AAh)
------------------------------------

-----------------------------------------------------
Partition entry (length 16 bytes)
-----------------------------------------------------
Off. Bytes Meaning

00h   1    80h = active partition / 00h = not active
01h   1    begin of partition (head number)
02h   1    begin of partition (sector number)   [*]
03h   1    begin of partition (cylinder number) [*]
04h   1    partition ID
05h   1    end of partition (head number)
06h   1    end of partition (sector number)     [*]
07h   1    end of partition (cylinder number)   [*]
08h   4    rel. sectors (# sec. to begin of partition)
0Ch   4    number of sectors in partition
-----------------------------------------------------

[*] Remark:
The byte for the sector number (offset 2 respectively 6) contains
the CHS sector number in the bits 0-5.
Hence in the CHS format the sector number occupies 6 bits.
The remaining two bits 6+7 belong to the CHS cylinder number,
they represent the high bits 8+9 of the CHS cylinder number.
The byte for the cylinder number (offset 3 respectively 7) contains
the low bits 0-7 of the CHS cylinder number.
Hence in the CHS format the cylinder number occupies 10 bits.
```

The byte at offset 0 specifies whether the partition is active or not. In the standard boot process (without boot manager) the partition marked as active is booted automatically.

Obviously that there are two ways to specify the position of a partition. Namely the CHS format and the LBA format. In the CHS format the position of a partition is specified by cylinder/head/sector numbers specifying begin and end of the partition (24 bit each). In the LBA format the position of a partition is specified by specifying the relative sector



(begin of partition) and total number of sectors (32 bit each).

For disks up to a size of **8 GB** the 24 bit of the CHS format are sufficient to specify all partitions.

For larger disks these 24 bits are not any more sufficient. The 32 bit of the LBA format allows addressing of all disk sectors up to a maximum disk size of **2048 GB** (2 TB).



6.4 Master Boot Record (MBR) / track 0

Master Boot Record (MBR):
The Master Boot Record (MBR) on x86−computers is the first sector of the hard disk. The MBR occupies exactly one sector (512 bytes). After starting the computer the BIOS automatically loads the first sector of the boot drive (C: or A:) and starts the code found on this sector. When the boot order is set to "C: , A:", then the BIOS tries to load the MBR from the first hard disk. When booting from A: comes first, then the BIOS tries to load the first sector of a diskette. When the sector has been read successfully (from C: or A:) it is executed. When the sector could not be read (e.g. booting from A: without diskette in drive A:) the BIOS tries to boot from the next boot drive.

On diskettes there is no MBR. Rather the first sector of the diskette contains the boot sector if the diskette is bootable, or a small program which asks the user to remove the diskette and press any key.

The MBR is a partition sector. All partition sectors have the same format (see format of partition sector), which is independent of the operating system. The MBR also contains a small boot loader program and the partition table. The partition table can contain up to four entries. Each entry describes a partition. The boot loader of the standard MBR searches in the partition table of the first hard disk for a primary partition marked as active. If such a partition is found, the boot sector of this partition is loaded and started. The boot sector finally loads the corresponding operating system.

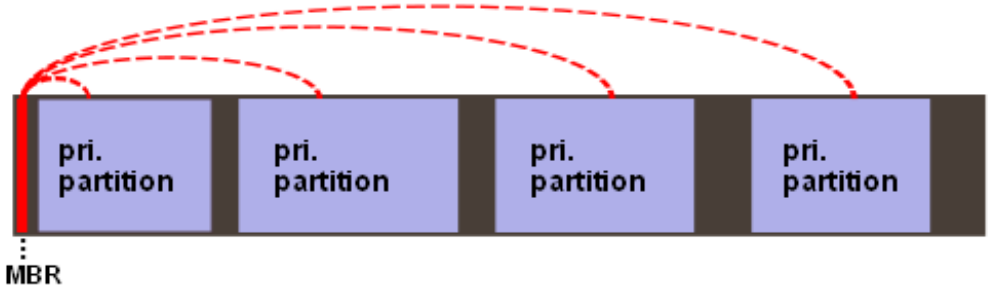
Track 0:
The term "track" denotes a set of sectors all having the same head number and cylinder number. The "track 0" consists of all sectors with head and cylinder number 0. The first sector in the track 0 is the MBR. Since the sector number is restricted to the range 1 to 63, the track 0 contains at most 63 sectors. Usually the track 0 is empty (except for the MBR), hence this space can be used store small programs. This has the advantage that no additional partition must be created to store these data and furthermore there is no dependency on the file systems of existing partitions. The problem is that there is no indicator whether the track 0 contains some data or not. Therefore it could happen that other programs unintentionally overwrite the data stored in the track 0.



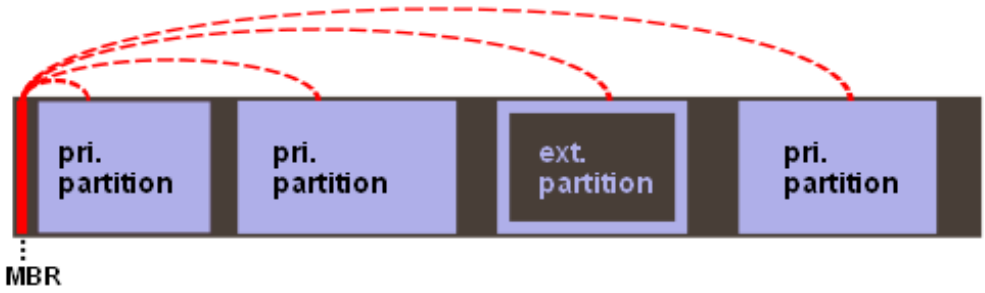
6.5 Primary / extended partition

The partition table in the partition sector of the MBR can contain at most four entries. This could be either only primary partitions (max. four) or exactly one extended partition and up to three primary partitions. This convention allows, for example, that on a second hard disk only one extended partition is created.

Primary partition:
A primary partition is completely defined by its entry in the partition table. This entry contains the information about the beginning and end of the partition, the type of the partition and the position of the accompanying boot sector.



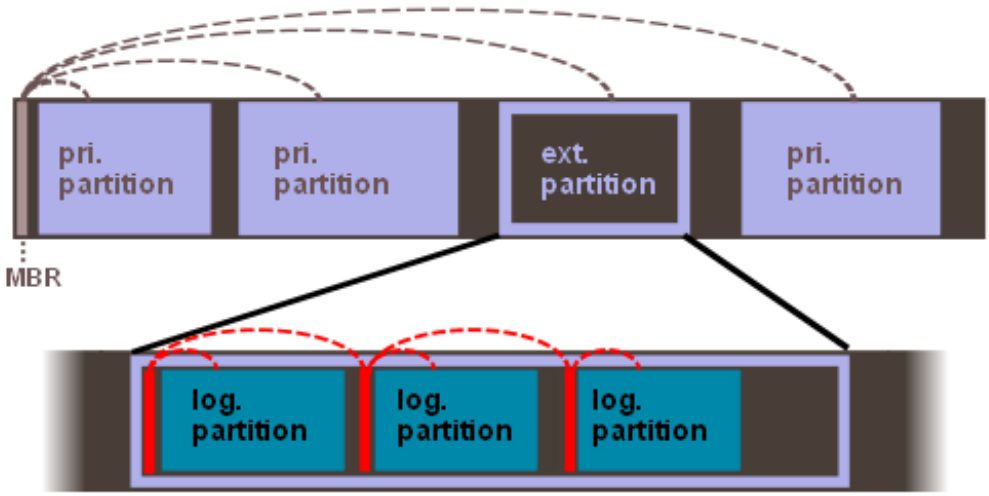
Extended partition:
In the case of an extended partition the entry in the partition table of the MBR describes only the space reserved for the extended partition. This space can be split in up to 24 logical drives. Logical drives can be created only inside an extended partition. In simple words, the extended partition represents the space reserved for the logical drives. The partition entry in the MBR points to the partition sector of the first logical partition, which resides directly at the beginning of the extended partition.



6.6 Logical drive

A logical drive –– often also called a logical partition –– is a set of adjoining sectors inside an extended partition. In many respects the logical drives resemble primary partitions. However the boot loader in the standard MBR of Microsoft does not support booting from logical drives. With the help of a boot manager a logical drive can be easily booted, supposed the operating system allows itself to be installed on a logical drive.

The size, position and type of logical drives is stored in a chain of partition sectors inside the extended partition. The partition table of a logical partition sector contains two entries. One entry describes the logical partition, the other entry contains the position of the partition sector of the next logical drive. The entry in the partition table of the MBR describes just the beginning of this chain. If the chain is broken, the following logical drives will not be accessible any more.



6.7 Active partition

The active partition is the partition on the first hard disk which is automatically booted when no boot manager is installed. The partition marked as active is set in the partition table of the MBR. Please note that there is only one active partition allowed on each hard disk. If a boot manager is installed it is irrelevant which partition is marked as active, since the partitions are selected and booted from the boot manager.



6.8 Hidden partition / true hidden partition

If more than one primary C: partitions are present on a hard disk (e.g. DOS 6.22 and Windows 95) the assignment of drive letters to these partitions might lead to problems. These problems can be avoided if all but one of these primary C: partitions are hidden. It is then possible to boot only from the remaining non–hidden partition. The other C: partitions are not visible, i.e. they are considered as unknown partitions. The drive letter C: is thus always automatically assigned to the booted operating system.

In order to hide partitions the value 10h is added to the partition ID. And in order to make a partition visible only this value 10h needs to be subtracted from the partition ID. The OS/2 boot manager uses the same scheme for hiding partitions.

Both in the configuration program Boot–US and in the boot manager Boot–US only partitions which use drive letters can be hidden. Hiding partitions therefore applies only to the following partition IDs: 01h, 04h, 06h, 07h, 0Bh, 0Ch and 0Eh, (see list of recognized partition IDs). All other partitions (e.g. Linux with partition ID 83h) cannot be hid.

True hidden partition:
The simple method of hiding a partition described above does not work with Windows 2000/XP/2003/Vista/2008/7/8. This means, even if the partition ID is switched to hidden, the partition is accessible under Windows 2000/XP/2003/Vista/2008/7/8. In order to **true hide** the partition the **boot sector** is additionally changed. Hence true hiding a partition means to change both the partition ID and the boot sector. Then the partition is not anymore recognized by Windows 2000/XP/2003/Vista/2008/7/8.

Comment:
It is possible that Windows 2000/XP/2003/Vista/2008/7/8 considers a true hidden partition as **unformatted**. This can be changed by **removing the drive letter** from a true hidden partition. This prevents that you accidentally format such an "unformatted" partition thereby deleting all existing data on that partition. Drive letters are assigned or removed within the Windows disk administrator (Computer Management / Storage / Disk Management).



6.9 Bootable partition

Boot−US considers partitions as **bootable** if the boot sector contains a valid **boot sector signature**. Specifically the last two bytes in the boot sector must be 55h AAh. When this signature is missing the corresponding partition is considered non−bootable. By this simple check Boot−US decides whether a primary or logical partition is bootable. Extended partitions are not bootable in general, since they represent only the frame for the logical partitions.

Unfortunately, the presence of this signature is **not a safe** indicator of a bootable partition. One could think that by checking the whole boot sector one could decide whether a partition is bootable or not. It would "only" be necessary to store a list of all boot sectors for all supported operating systems and compare the actual boot sector to that list. However, this would mean a big effort and even then it would not allow a safe identification of bootable partitions. Even if the boot sector is fully correct the boot process can fail in a later state, for example when a driver is missing or does not fit to the hardware. Probably by now many user might have experienced this by themselves.

Therefore in practice the bootability of a partition can be verified only by **experiment**, i.e. by trying to boot the partition. If it works the partition is bootable, otherwise not. Thus partitions which are classified as bootable by Boot−US could also turn out as **non−bootable** in reality.



6.10 Which partition IDs are recognized ?

Boot–US knows 70 partition IDs in total. The following table contains the list of these partition IDs with a short description of the corresponding partition type. Please note that some partition IDs cannot be assigned uniquely to a specific partition type. In other words several operating systems respective partitions use the same partition ID.

Partition ID	Partition type
00h	partition is free (empty)
01h	FAT12
02h	Xenix root
03h	Xenix /usr
04h	FAT16 < 32 MiB
05h	extended partition
06h	FAT16 > 32 MiB
07h	IFS (HPFS, NTFS)
08h	OS/2, AIX, Dell, QNX
09h	AIX, Coherent, QNX
0Ah	OS/2 boot manager
0Bh	FAT32
0Ch	FAT32 (LBA)
0Eh	FAT16 (LBA)
0Fh	extended partition (LBA)
10h	OPUS
11h	FAT12 (hidden)
12h	Compaq config partition
14h	FAT16 < 32 MiB (hidden)
15h	extended partition (hidden)
16h	FAT16 > 32 MiB (hidden)
17h	IFS (HPFS, NTFS) (hidden)
18h	AST
1Bh	FAT32 (hidden)
1Ch	FAT32 (LBA) (hidden)
1Eh	FAT16 (LBA) (hidden)
1Fh	extended partition (LBA) (hidden)
24h	NEC DOS 3.x
38h	THEOS ver 3.2
39h	THEOS ver 4
3Ah	THEOS ver 4
3Bh	THEOS ver 4 extended partition
3Ch	Partition Magic recovery partition
40h	Venix 80286
41h	Personal RISC
42h	SFS
44h	GoBack partition
45h	Boot–US boot manager
4Dh	QNX 4.x
4Eh	QNX 4.x 2nd part
4Fh	QNX 4.x 3rd part
50h	OnTrack DiskManager RO
51h	OnTrack RW, Novell
52h	CP/M, Microport
53h	Disk Manager 6.0
54h	Disk Manager 6.0
55h	EZ-Drive
56h	GoldenBow
61h	SpeedStor
63h	Unix System V



```
      64h      Novell Netware 2.xx
      65h      Novell Netware 3.xx, 4.xx
      67h      Novell
      68h      Novell
      69h      Novell
      70h      DiscSecure MultiBoot
      75h      IBM PC/IX
      80h      Minix up to 1.4a
      81h      Minix from 1.4b, Linux (early ver.)
      82h      Linux swap
      83h      Linux native (usually ext2fs)
      84h      Hibernation partition
      86h      FAT16 volume set
      87h      NTFS volume set
      A0h      Phoenix Save-To-Disk
      A1h      Phoenix Save-To-Disk
      A5h      BSD
      A6h      OpenBSD
      A9h      NetBSD
      B6h      Windows NT FAT16 mirror set (corrupt)
      B7h      Windows NT NTFS mirror set (corrupt)
      BEh      Solaris 8 boot partition
      BFh      New Solaris x86 partition
      C6h      Windows NT FAT16 volume set (corrupt)
      C7h      Windows NT NTFS volume set (corrupt)
      DBh      Concurrent DOS
      EBh      BeOS
      ECh      SkyOS
      EEh      EFI header
      EFh      EFI file system
      FFh      Bad Track Table
-------------------------------------------------
```

Some partitions are classified as hidden partition. The partition ID of the hidden partitions is different from their visible counterpart only by the hex number 10h which is added on the partition ID of the visible partition.



6.11 LBA and CHS format, LBA mapping

LBA format:
LBA is the abbreviation for Logical Block Address. This simply means the sectors of a disk are sequentially numbered starting with LBA number 0. Every sector is identified by its unambigious LBA number. The LBA format is supported only by a mainboard BIOS which is not too old.

CHS format:
In the CHS format a sector is identified by its Cylinder/Head/Sector number. Earlier these values meant the physical location of the sector on the disk. Today these values are only logical numbers. The (logical) number of cylinders, heads and sectors per track can be determined by the func. 08h respectively 48h of the BIOS interrupt 13h. The CHS format is supported by all the BIOS but it is limited to a disk size of **8 GB** (see below).

The CHS format has the following restrictions:

```
cylinder number    0 - 1023    (10 Bit)
head number        0 - 254      (8 Bit)
sector number      1 - 63       (6 Bit)
```

Due to a problem in the DOS operating system a maximum head number 255 leads to a crash although the BIOS would allow this number. Therefore a maximum head number 255 is not used in general.

Max. disk size:
For the CHS format in the partition table and for calling BIOS funtions there are in general only 24 bits available. With 24 bits and a sector size of 512 bytes it is possible to address all sectors of a disk up to a maximum disk size of **8 GB**. For the LBA addressing scheme there are 32 bits reserved in the partition table. At the same sector size of 512 bytes all sectors of a disk up to a maximum size of **2048 GB** (2 TB) can be addressed. The extended INT 13h even uses 64 bits for storing the LBA number. Due to its simpler concept and also the possibility to support larger disks the LBA format is used internally by all modern operating systems for the direct disk access.

Conversion between CHS and LBA:
It is possible to covert the LBA format to CHS and vice versa. Conceptually both forms are equivalent. A sector C/H/S in the CHS format has the following LBA number:

```
LBA = C x Num_Head x Num_Sec + H x Num_Sec + (S - 1)
```

Here `Num_Sec` means the (logical) number of sectors per track and `Num_Head` the (logical) number of heads. Only these two (logical) geometry parameters of the disk are relevant for the conversion. The number of cylinders in unimportant for the conversion.

LBA mapping:
The two conversion parameters `Num_Head` and `Num_Sec` are summarized as LBA mapping. The LBA mapping determines the conversion between CHS and LBA numbers of a sector. These two parameters are obtained from the function AH=08h of the standard BIOS interrupt 13h. Even if the extended interrupt 13h is supported, the LBA mapping is obtained from the standard INT 13h. The disk parameters which are returned from the extended INT 13h are useful only to compute the total disk size.

The total disk size is simply the product of the total number of sectors (number of cylinders `x` number of heads `x` number of sectors/track) with 512 (the sector size).



6.12 INT 13h / extended INT 13h

INT 13h:
The BIOS interrupt 13h provides functions for direct access of the disk. These functions are implemented by the BIOS, i.e. these functions can be used without the need of any disk drivers. This is important especially for booting, since at this moment no operating system and no hard disk drivers are loaded. Typically the INT 13h is used for reading the boot sector from the disk. The INT 13h uses the CHS notation with a width of 24 bit in order to address a sector. Therefore the INT 13h can access at most **8 GB** of a disk. If the disk is larger, then in the best case the first 8 GB are accessible by the INT 13h.

Extended INT 13h:
The extended INT 13h is the extension of the INT 13h. This extension has been specified at the beginning of the Nineties by IBM and Microsoft. One difference (among others) of the extended INT 13h is that it uses LBA numbers with a width of 64 bit for addressing disk sectors. Therefore the extended INT 13h supports disks up to a size of **8 billion TB**. This limit exceeds today's typical disk capacities (8 GB) by 12 orders of magnitude. Hence it should be possible to use the extended INT 13h for "some" more years.

Basically the configuration program Boot−US uses "only" 32 bit for the LBA numbers. Please note, however, that also the partition table uses LBA numbers with a width of 32 bit. Even these 32 bit still allow to address all sectors up to a disk size of **2048 GB** (2 TB).





7 Tech–Info

In the following sections the basic structure and design of the configuration program Boot–US and the corresponding boot manager is explained.

1. Structure of the configuration program Boot–US
2. Structure of the boot manager of Boot–US
3. Standard boot process
4. Boot process with boot manager in primary partition
5. Boot process with boot manager in MBR
6. Boot process with boot manager on diskette
7. Booting from "second" disk
8. Uninstalling the boot manager
9. Partition numbers



7.1 Structure of the configuration program Boot−US

The configuration program Boot−US determines all partitions on all local hard disks as soon as it starts. This information is obtained by analysing the partition tables of all partition sectors on each local hard disk. Then all partitions found are displayed with some important data. This data is obtained by Boot−US from the standard partition tables Boot−US recognizes and handles all operating system partitions. Hence Boot−US is not restricted only to DOS or Windows partitions. When new partitions are created, changed or deleted all modifications are automatically visible at the next startup of Boot−US or after a manual refresh of the display. It is not necessary to restart the PC to see the changes.

When the configuration program Boot−US is started, no data on any disk sectors are changed. Disk sectors are written only when the boot manager Boot−US is installed or when the partition tables are modified.

Inside Boot−US all read or write access to disk sectors happens directly under Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8. Switching to DOS mode is not required at all. Especially, with Windows 95/98/ME direct disk access is not easy to achieve. There are limited number of programs which are able to do that. As a result, reading the partition tables and potential modifications happen in the same program (Boot−US). Additionally, the Windows 95/98/ME and NT/2000/XP/2003/Vista/2008/7/8 version of Boot−US are identical, except for the code for direct disk access. All these measures increase the stability of Boot−US and minimize the number of errors.

The configuration program Boot−US uses internally only the LBA format for addressing the disk sectors. All high−level operations employ the LBA format. CHS sector numbers are used only for the low−level direct disk access under Windows 95/98/ME when the extended INT 13h is not supported. In this case the LBA sectors numbers are converted to the CHS format before the disk sectors are accessed by the low−level INT 13h functions.



7.2 Structure of the boot manager Boot−US

The boot manager Boot−US is a small assembler program with a total size of about 10 KB. The complete code of the boot manager is included in the configuration program Boot−US. It is not delivered in a separate file. The boot manager is installed and removed from the disk (or diskette) by this configuration program.

Internal structure:
Basically the boot manager consists of two parts called **LOADR−US** and **BMGR−US**.

LOADR−US is the smallest part of the boot manager. LOADR−US has a size of about 200 Bytes. This part can be stored completely on a single disk sector (512 Byte). The task of LOADR−US is to load and start the main part BMGR−US from disk (or diskette). Although the boot manager occupies many sectors, it could be started by loading a single sector (LOADR−US). Hence conceptually LOADR−US is similar to a boot sector (see standard boot process).

The actual boot manager is contained in the **main part BMGR−US**. BMGR−US occupies about 20 disk sectors. BMGR−US shows a menu containing all partitions included in the boot manager and waits for the user to select one of them. When a partition is selected the boot sector of this partition is loaded from the corresponding disk(or diskette) and executed. Hence the selected partition is booted.

Disk size up to 2048 GB:
When the BIOS supports the extended INT 13h then inside of BMGR−US the LBA format together with the extended INT 13h is used for loading the boot sector of the selected partition, supposed the boot sector is beyond the 8 GB limit. This allows the boot manager Boot−US to access disks up to a size of 2048 GB. In older PCs the BIOS does not contain support for the extended INT 13h. In such cases BMGR−US can only use the standard INT 13h and then the boot partition must reside below the 8 GB limit.

Modern operating systems like Windows ME/2000/XP/2003/Vista/2008/7/8 can now−a−days be installed also in partitions beyond the 8 GB limit. The boot manager Boot−US has no problems to boot these (and other) operating systems beyond the 8 GB limit (assuming that the BIOS supports the ext. INT 13h).

Comment:
At the time of installation of the boot manager all necessary data of the configured partitions are stored in the main part BMGR−US of the boot manager. Especially the disk number and the position of the boot sector of each configured partition is stored there. When the beginning of one of those partitions is changed, for example when the partition is moved, the boot manager cannot find the boot sector any more. In such a case the boot manager must be installed again. Also when the disk number changes (by adding/removing a hard disk) the boot manager must be reinstalled.



7.3 Standard boot process

The standard boot process (without boot manager) proceeds as follows: After starting the computer the BIOS automatically tries to read the first sector of the boot device (C: or A:). When this succeeds the code in that sector is executed. In case of an read error (eg. booting from A: without diskette in drive) the BIOS tries to boot from the next boot device.

Hence when booting from C: the BIOS reads the first sector −−− the Master Boot Record (MBR) −−− from the first hard disk and starts the boot loader code found on this sector. This small program (size 200 − 300 bytes) searches the partition table for the active partition. The partition entry contains the position of the corresponding boot sector. When an active partition is found then the boot loader code reads the boot sector (from the first hard disk) and executes the code contained on the boot sector. This code finally boots the corresponding operating system.

Since the standard boot loader searches only in the partition table on the first hard disk, it allows booting an operating system only from the first hard disk. Also booting from a logical drive is not possible with the standard boot loader.

On the diskette there is no MBR. Rather the first sector on a diskette already contains the boot sector of an operating system in case the diskette is bootable, or the well−known little program which asks the user to remove the diskette and press any key.

General principle:
The load process outlined above is an example of a widely used general principle. In the first step only a single sector is loaded and executed. This sector contains the necessary code to load additional sectors. Only these sectors loaded in the second step contain the actual program to be executed. In this way a big program (e.g. an operating system) can be started simply be loading a single sector (e.g. a boot sector).



7.4 Boot process with boot manager in primary partition

There are two possibilities:

Boot manager partition is permanent active

This possibility is present in all previous versions of Boot−US. Initially the boot process is identical to the standard boot process. Thus the BIOS loads the MBR which searches for an active partition and loads the corresponding boot sector.

The active primary partition on the first hard disk is that partition in which the boot manager of Boot−US is installed. This partition must be permanetly active. The first sector of this partition contains the first part (LOADR−US) of the boot manager. The task of LOADR−US is −−− similar to a boot sector −−− to load the main part (BMGR−US) of the boot manager and start it.

BMGR−US shows a menu with all configured partitions and waits for the user to select a particular partition. Then BMGR−US loads the boot sector of the selected partition from one of the local hard disks or from the diskette and executes the code found on the boot sector. In this way the selected partition is finally booted.

Boot manager partition is not active

This possibility is implemented the first time with Boot−US 3.0.0. Technically the boot process is similar to the boot process with boot manager in MBR. The BIOS loads the MBR which contains now the first part of the boot manager (LOADR−US). This code from the MBR loads directly the first sector of the boot manager partition, which in turn loads the complete boot manager. The contents of the boot manager partition is identical in both cases. Only the code in the MBR is different. In the second case it does not search for the active partition instead it directly loads the boot manager partition.



7.5 Boot process with boot manager in MBR

When the boot manager of Boot−US is installed in the MBR, the boot manager becomes active in an initial stage of the boot process. When the computer is restarted the MBR of the first hard disk is loaded and executed. This procedure is still identical to the standard boot process. However the boot loader code of the MBR now does not contain the usual boot loader but the first part (LOADR−US) of the boot manager Boot−US. When executed, LOADR−US loads the main part of the boot manager (BMGR−US) und starts this main part (BMGR−US).

From this moment the boot process is identical to the boot process where the boot manager is installed on a primary partition. Thus the user selects one of the configured partitions and the boot manager loads and executes the boot sector of this partition. This will boot the corresponding operating system.



7.6 Boot process with boot manager on diskette

When a diskette is booted the first sector of the diskette is read and executed. Normally this sector would be the boot sector of an operating system. Now the first sector of the diskette contains the first part (LOADR−US) of the boot manager Boot−US. When executed, LOADR−US loads the main part of the boot manager (BMGR−US) from diskette and starts this main part (BMGR−US).

From this moment the boot process is identical to the boot process where the boot manager is installed on a primary partition. Thus the user selects one of the configured partitions and the boot manager loads and executes the boot sector of this partition. This will boot the corresponding partition.



7.7 Booting from "second" disk

Practical problems
When an operating system is installed on a particular hard disk and later new hard disks are installed on the PC, then it may happen that the **disk number** of the existing disks **might change** −− intentionally or unintentionally. For example it would be reasonable to install a new (faster) disk as the first disk and use that disk during normal operation. On the other hand one would often like to still use the existing operating systems on the old hard disks. In the case of the Microsoft operating systems (DOS and Windows) things are complicated by the fact that they can be installed easily only to the **first** hard disk. Thus in practice one often deactivates the existing (old) disk so that the target disk appears as the first disk during the installation. Without boot manager multiple Windows partitions on different disks are then used with one of the following typical methods: the disturbing disks are turned off at the BIOS level or their disk number is always changed manually or they are even completely turned off at the hardware level. None of these "solutions" is really convenient. Similar problems exist also with non−Microsoft operating systems. It is not often seen that an operating system can still be booted when the disk number of the boot partition has **changed**.

Solution in boot manager Boot−US
Technically speaking the problem is that in the boot partition at certain positions the disk number is stored. When the disk number is changed this information becomes invalid and the partition cannot be booted any more. It depends on the operating system whether the disk number is stored only in the boot sector or also in other system files.

In the case of **DOS**, **Windows 95/98/ME** it seems that indeed the disk number is stored only in the boot sector and in no other system files. In order to boot e.g. Windows 95 from a disk which is currently the second disk while it was the first disk at installation time, one needs to insert "only" the current disk number in the appropriate positions of the boot sector and then start this modified boot sector. Actually it is even sufficient to change the boot sector on the fly in memory. The boot sector on the hard disk remains unchanged. The **boot manager Boot−US** uses this method for booting DOS and Windows 95/98/ME from another (second) hard disk.

Starting with Boot−US 2.0.0 it is possible to boot **Windows NT/2000/XP/2003/Vista/2008/7/8** from the second, third,... hard disk. From a technical point of view this is achieved by swapping the drives for the BIOS interrupt 13h. This swapping is visible only on the BIOS level, i.e. during the boot phase. After Windows NT/2000/XP/2003/Vista/2008/7/8 is running all further disk access does not use the BIOS interrupt 13h any more but the Windows disk drivers. Although the disk swapping is still present on the BIOS level it does not influence the operation of Windows.

These explanations make clear that **no general** solution for this boot problem exists. Each operating system and partition requires separate consideration. Boot−US contains explicit support for booting from another disk only for **DOS** and **Windows 95/98/ME/NT/2000/XP/2003/Vista/2008/7/8**.

What about other operating systems ?
The boot manager of Boot−US passes the current disk number in the DL register. This might be sufficient for non−Microsoft operating systems to boot them from the corresponding hard disk. However it must be tried out whether this actually works or not.



7.8 Uninstalling the boot manager Boot−US

During the installation of the boot manager Boot−US the original MBR was saved either on the boot disk 1 or on diskette.

In case the boot manager Boot−US had been installed on a primary partition and the boot manager is removed, then the boot manager partition is deleted and the former active partition is re−activated. The information about the previous active partition is obtained from the saved MBR. If necessary, the first partition is activated. The boot loader code of the MBR has not been changed when the boot manager was installed and therefore does not need to be restored.

In case the boot manager Boot−US is installed in the MBR and the boot manager is removed, the previous boot loader code must be restored on the MBR. This code is copied from the saved MBR. Additionally the former active partition is searched in the previously saved MBR and reactivated in the current MBR.

The menu command **Boot manager / Remove...** first checks the current installation of the boot manager and then performs the appropriate type of uninstallation. Normally it is not necessary to restore the complete track 0.

The commands **bootusc bootmanager remove** in the command−line version and **Uninstall boot manager** in the boot manager perform exactly the same operations as the uninstallation from the GUI.

Detailed technical description of uninstallation:

- The current MBR of the boot disk 1 is read.
- When the boot manager is installed on the MBR the previously saved MBR is read from the second sector of the boot disk 1 (LBA=1), and the MBR code of the current MBR is replaced by the MBR code of the previously saved MBR.
- When the boot manager is installed on a primary partition the previously saved MBR is read from the second sector of this boot manager partition or from the second second sector of the track 0. The MBR code in the current MBR remains unchanged.
- The former active partition is searched in the previously saved MBR.
- A potentially existing boot manager partition is deleted.
- The former active partition in the current MBR is reactivated.
- The partition table of the current MBR remains unchanged otherwise.
- The modified current MBR is written back to the boot disk 1.
- All partitions on all disks are unhid.

By the above operations the MBR code is restored (only if necessary) and a potential boot manager partition is removed. Effectively the boot manager is uninstalled and the previous state is restored.



7.9 Partition numbers

Within the package Boot−US all partitions on a hard disk are assigned a partition number according to the following rule:

Primary and extended partition are assigned always a partition number between **1** and **4**. Logical partitions are assigned always a partition number of **5** and higher.

Within each of the two groups the partitions are ordered according to their position on the hard disk. For example a primary (or extended) partition with the number 1 lies before a partition with the number 2. Similarly a logical partition with the number 5 always lies before a log. partition with the number 6.

