

Index

USB Drive Letter Manager V5.2.5

[Getting Started](#)

[Installation](#)

[Configuration](#)

[Problems](#)

[History](#)

[Licence agreement](#)

Homepage:

http://www.uwe-sieber.de/usbdlm_e.html

General tips for USB drives:

http://www.uwe-sieber.de/usbstick_e.html

EMail:

mail@uwe-sieber.de

Getting started

USB DLM is a Windows service that gives control over Windows's drive letter assignment. Initially made for USB drives it can now control drive letters of other drive types too.

Running as service makes it independent of the logged on user's privileges, so there is no need to give the users the privilege to change drive letters.

It automatically solves conflicts between USB drives and network or subst drives of the currently logged on user.

Furthermore you can define new default letters for USB drives, reserve drive letters and much more.

It can deal with non USB drives too when a [BusType](#) or [DeviceID](#) configured.

USB DLM works with XP, Vista, Windows 7, 8, 8.1, 10 and all the Windows Server versions. The x64 version works on x64 editions of Windows only.

Windows 2000 support has been dropped with V4.8.

It is primarily tested under

- Windows XP SP3
- Windows 7 x64
- Windows 10

USB DLM is not compatible with Microsoft Multipoint Server clients.

Installation

USB DLM does not come with a SETUP.EXE, it's all manual work. See Section [Installation](#).

An MSI is available which installs to %ProgramFiles%\USB DLM. And you can [build your own MSI](#).

Conditions

The software is not limited or crippled in any way. It's fully functional without a time limit. There is no "full version".

USB DLM is Freeware for private and educational use only. Otherwise a licence is required per computer after a 30 day test period.

"Educational" means institutions where the students outnumber the employees.

Furthermore it's free for public, free of charge libraries.

Price list is available here:

http://www.uwe-sieber.de/usbdm_e.html

[Licence agreement](#)

Why?

When a drive (e.g. a USB flash drive) is attached for the first time, Windows mounts it to the first available drive letter. The drive letter can be changed in the Windows Disk Management console, but Windows can save only one drive letter per drive and vice versa. So, you cannot have more than one drive on a high drive letter **in succession**.

With USBDLM you can

- check if the letter is used by a network share of the currently logged on user and assign the next letter that is really available (no configuration required)
- reserve letters, so they are not used for local drives
- assign a letter from a list of new default letters, also dependent on many different criteria as the active user, drive type, type of connection (USB, FireWire), USB port, volume label, size, user and others
- let remove the drive letters of card readers as long as there is no media present
- ask the user for a password before he can use a drive
- let show an balloontip on drive arrival which shows the assigned drive letter
- run something on click on the balloontip
- executing an autorun, also depending on the criterions mentioned above
- and much more...

All functions are applied to USB drives at the moment they are being attached, when the USBDLM service starts up and when a user logs on.

Instead of letters you can set up empty folders on NTFS drives as mount points too but this brings some limitations and problems, see [configuration](#).

History

V5.2.5 (31 Aug 2016)

- Bugfix: opening VBS files USBDLM did not pass parameters
- New: USB Drive Info shows problem codes from the device manager in the tree nodes
- New: Autorun events [OnSafeRemoval](#) and [OnUnsafeRemoval](#)

V5.2.4 (22 June 2016)

- Bugfix: Since V5.1.7 [OnDeviceArrival] did not work
- Bugfix: SuppressWindowsBalloons=1 did not work on non-english Windows 10
- Bugfix: Ballontip's fadeout effect was broken under Vista/Win7,8,10 with non-classic display
- Bugfix: open= did not work with a bat or cmd file in quotes
- New: Under Windows 10 with SuppressWindowsBalloons=0 USBDLM's balloons for volumes with a drive letter are delayed by 3 Seconds to make the Windows 10 balloon come first; The delay [can be changed](#)
- New: Search function in UsbDriveInfo

V5.2.3 (5 June 2016)

- Bugfix: Since V5.2.2 creating USBDLM.LOG backups did not work with LogLevel >= 3
- New: In the [log file](#) the date can be written in front of each line (LogShowDate=1)
- New: In the log file timing, loglevel and thread ID can be omitted (LogShowTech=0)

V5.2.2 (28 May 2016)

- Bugfix: open= added the executable twice instead once as parameter under Vista, Win7,8,10
- New: USBDLM service start/stop a event in the [EventLog](#)
- New: Number of USBDLM.LOG backups can be [set](#)

V5.2.1 (10 May 2016)

- New: Unwanted [EventLog](#) items can be deactivated

V5.2.0 (8 May 2016)

- Bugfix: USBDLM_usr.exe (still) sometimes caused permanent CPU load
- Bugfix: Since V5.0 USBDLM has set WriteCacheEnableOverride=1 into the Registry for newly attached USB drives if there was no UsbWriteCacheEnable configured
- New: [EventLog](#) can be written to the LogFile instead

V5.1.8 (25 April 2016)

- Bugfix: USBDLM_usr.exe sometimes caused permanent CPU load, probably since V5.0

- Bugfix: open= did not work with document files since V5.1.3
- Bugfix: USB speed icon not shown on some USB controllers running with other than Microsoft drivers
- Bugfix: Maximum connection speed of USB 2.0 High-Speed devices running at Full-Speed shown wrong as USB 1.1 Full-Speed

V5.1.7 (07 April 2016)

- Bugfix: USB speed icon not shown on some USB controllers not running with Microsoft drivers
- Bugfix: Since V5.1.5 the PortName criteria did not work again for USB drives
- Bugfix: Several minor issues fixed
- New: OnArrival is triggered on instertion of a media into a CD/DVD drive

V5.1.6 (17 March 2016)

- Bugfix: Since V5.0 USBDLM stooped assigning drive letters after attaching a drive with a BusType for which there is no section DriveLetters

V5.1.5 (11 March 2016)

- Bugfix: In V5.1.4 crash on balloon tip with NTFS mountpoint
- Bugfix: Since V5.0 settings in the registry did not work

V5.1.4 (09 March 2016)

- Bugfix: In V5.1.3 the Win32 USBDLM.exe was compiled without optimization for "Minimize Size", besides the 100KB bigger USBDLM.exe this made some virus scanners believing to detect a virus
- Bugfix: Since V5.0 the PortName criteria did not work for USB drives
- Rollback: Default for [SuppressWindowsBalloons](#) is 1 again; 2 (deactivating Windows balloon tips temporarily) makes the Windows tray icon for "Safely remove Hardware" stop working

V5.1.3 (07 March 2016)

- Bugfix: Occcational crash fixed

V5.1.2 (03 March 2016)

- Bugfix: Logfile created even WriteLogfile=0

V5.1.1 (29 Feb 2016)

- Bugfix: AutoDismount-TrayIcon was shown even there is no [AutoDismount] section

V5.1.0 (29 Feb 2016)

- Bugfix: In sections as [OnArrival] the lines system, elevated and restricted where always resolved as 1 if present
- Bugfix: In sections as [OnArrival] with only one open= line options set before the open line had no effect
- New: Tray icon for showing the [Auto-Dismount](#) status

- New: Flush and DismountOnSleep does not wake up sleeping drives

V5.0.2 (19 Jan 2016)

- Bugfix: In a USBDLM.INI on an attached drive only Letter= and Letters= worked, but Letter1= etc possibly not

V5.0.1 (18 Jan 2016)

- Bugfix: _edit-ini.cmd failed to create a USBDLM.INI when it had no write privileges; now Admin privileges are requested
- Bugfix: USB speed in the balloon tip was wrong when the USB port number could not be determined, for instance if in the location information there isn't something like Port_#0002.Hub_#0003 or the address value is zero. Now the port number is determined by means of the USB-API.
- Bugfix: The section [NetworkLetters] did not work in V5.0.0 if there was not section [DriveLetters]

V5.0.0 (17 Jan 2016)

- New: _install.cmd and _uninstall.cmd renamed to what they really do: _service_register.cmd and _service_deregister.cmd; _start.cmd and stop.cmd renamed to _service_start.cmd and _service_stop.cmd to fit in the new naming scheme
- New: For sections and values there is no more need for numbering; For unnumbered sections the order of appearance is what counts
- New: Ballontips show in the icon if a USB drive is running at Full-, High-Speed, SuperSpeed or SuperSpeedPlus
- New: [Ballontips](#) on removal with information if the removal was safe or not (still need translations for new texts)
- New: [Compact balloon tips](#) for multi-slot card readers if only one slot has a media present
- New: Balloontips can be [customized](#) way more flexible
- New: [AutoDismount](#)
- New: Suppressing Windows Balloontips works under Windows 10
- Bugfix: After a successful request for a password sometimes no drive letter was assigned
- Bugfix: Removal Policy was written to the registry even no OptimizeUsbDrivesForSpeed parameter was configured

V4.9.1.0 (30 Aug 2015)

- Bugfix: starting cmd and bat files with spaces in the path did not work since V4.8.8.6

V4.9.0.0 (04 Aug 2015)

- Bugfix: DeviceTypes CardReader and MSCR not set when the Disk device has the DriveType FIXED
- New: Suppressing Windows balloon tips works with Windows 10

V4.8.9.9 (12 July 2015)

- Workaround: Windows 7's Alt-Tab facility with Aero enabled is [screwed up](#) if there is a visible zero sized topmost window around; USBDLM used one for receiving volume notifications; now it is no more topmost nor visible which seems to make Windows 7 happy
- Workaround: DismountOnSleep=1 caused a 10 Seconds delay when USBDLM tries to dismount the drive-letter-less "System Restore" partition for an unknown reason; USBDLM doesn't try to dismount this volume anymore

V4.8.9.8 (15 June 2015)

- Bugfix: PnP Manager errors in Windows XP's EventLog because USBDLM unregistered devicenotifications too late on service stop
- Bugfix: With Windows AutoMount disabled USBDLM did not bring volumes online even there is a valid DriveLetters section for it
- Bugfix: MD5 check for open= worked only with uppercase MD5 hashes

V4.8.9.7 (10 May 2015)

- Bugfix: Drive letter of iPods and ignored devices was removed by USBDLM since V4.8.6

V4.8.9.6 (29 March 2015)

- Bugfix: UsbWriteProtect did not work if the registry key HKLM\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies doesn't exist

V4.8.9.5 (09 March 2015)

- New: Limits of AutoRun delay, wait etc. increased to 24h; time units can be used, e.g. 10s instead of 10000
- New: [Letters by Time](#) can be defined using multiple times
- New: [Letters by Time](#) can be defined depending on the weekday

V4.8.9.4 (15 February 2015)

- Bugfix: Activation of potential BadUSB devices didn't always work
- Bugfix: Windows since Vista with active UAC triggers a volume arrival event for TrueCrypt volume long after they had been mounted sometimes; Now USBDLM skips Balloontip and AutoRun here

V4.8.9.3 (02 February 2015)

- Bugfix: Since V4.6.9.2 crash USBDLM when executing system event

V4.8.9.2 (01 February 2015)

- Bugfix: Occasional crash USBDLM on arrival of a TrueCrypt volume

V4.8.9.1 (18 January 2015)

- Bugfix: Occasional crash USBDLM on arrival of a volume
- Bugfix: Removed access to generic floppy drives on USBDLM startup under XP

V4.8.9.0 (28 December 2014)

- Bugfix: Since V4.8.4 virtual volumes not handled correctly under Vista+ with active UAC
- Bugfix: Drive size in balloon tips now rounded to full Gigabytes again
- New: Support for the TrueCrypt replacement [VeraCrypt](#)
- New: UsbDriveInfo can add and remove NTFS mountpoints

V4.8.8.8 (10 November 2014)

- Bugfix: On creation of the initial USBDLM.INI the line NoMediaNoLetter=0 was written twice
- New: [DeleteSviFolder](#) works under other Windows versions than XP

V4.8.8.7 (14 October 2014)

- Bugfix: Keep Alive sectionname was documented wrong as DriveKeepAlive, correct is DiskKeepAlive
- Bugfix: USBDLM sometimes created unnecessary no-letter entries in the MountManager database

V4.8.8.6 (12 October 2014)

- Bugfix: On AutoRun of CMD and BAT files (e.g. open="C:\Batch\test.cmd") no params passed since V4.8.8.0
- Bugfix: Holding the Shift key pressed to skip AutoRuns did not work since Vista; "Solution": Force=1 is default now under Vista+ (Shift key ignored) and USBDLM_usr.exe is asked for the Shift key state if Force=0 is set for an open line
- Bugfix: UsbDriveInfo x64 showed several debug information not intended for end users
- Bugfix: Balloontip text for multiple partitions was used even there is only one partition with a volume
- New: [Keep Alive](#) for drives

V4.8.8.5 (19 September 2014)

- Bugfix: In some situations no letter was assigned, no autoruns executed and balloon tips not shown
- Bugfix: Multiple FIXED drives attached at the same controller no more shown in a common balloon tip

V4.8.8.4 (17 September 2014)

- Bugfix: In some situations autoruns wasn't executed and balloon tips not shown

V4.8.8.3 (27 August 2014)

- Bugfix: Crash on very long volume kernelnames

V4.8.8.2 (21 August 2014)

- Bugfix: Handling of potential "[BadUSB](#)" devices did not work under Vista+ unless a [OnDeviceArrival] section is present
- Bugfix: Since V4.6 letters in [NetworkLetters] did not overrule letters in [DriveLetters] sections (as documented)

V4.8.8.1 (19 August 2014)

- Bugfix: Since V4.8.8.0 the logfile was always written to C:_USB DLM.LOG
- Bugfix: UsbDriveInfo did not read USB DLM.INI changes on refresh

V4.8.8.0 (17 August 2014)

- Bugfix: Since V4.8.7 the USB port criteria was not reliable
- Bugfix: Since V4.8.7 newly attached volumes not handled correctly if there was another volume with an identical device ID attached before
- Bugfix: Since V4.8.7 on arrival of a drive the drive letter was removed for a short moment even if it was not required
- Bugfix: Since V4.5.9 OnUserLogon section executed for the first user only logged on at the same time
- Bugfix: When registering as Windows service by means of the Windows API function CreateService USB DLM passed the path to the USB DLM.exe in the parameter lpBinaryPathName unquoted even if it contains spaces. Despite the name of the parameter the Service Control Manager resolves it as a command line, thus with C:\Program Files\USB DLM\USB DLM.exe it would execute an existing C:\Program.exe with system privileges. The MSI did it right. USB DLM corrects the path without a reinstall.
- New: [Setting](#) SafeCommandLines to reject ambiguous open= command lines
- New: UsbDriveInfo warns for command lines which contain spaces but the executable not quoted
- New: "[BadUSB](#)" blocking: On arrival of an USB keyboard or network device USB DLM can ask the user if this is ok

V4.8.7.0 (29 June 2014)

- Bugfix: Since V4.8.0 crash USB DLM on logout when a virtual volume (Subst, Network or TrueCrypt drive) is present and on some other occasions
- New: UsbDriveInfo keeps device which are prepared for safe removal in the list
- New: UsbDriveInfo detect media changes by means of Windows notification messages instead of polling once a second
- New: Under Windows 7 and higher UsbDriveInfo can attach and detach VHDs, under XP and Vista if VHDmount from Microsoft Virtual Server 2005 is installed in its default folder, thus the vhdmount.dll is found in %ProgramFiles%\Microsoft Virtual Server\Vhdmount

V4.8.6.0 (24 April 2014)

- Bugfix: Several causes for crashes since V4.8.0 fixed
- New: USB DLM variable %DateISO%, e.g. 2014-04-14
- New: USB DLM tries on arrival of a disk device to give the mount manager data for the volume. This prevents Windows from assigning a drive letter which USB DLM then has to change. Since Windows 8 this does not work because it assigns the drive letter too fast. At least USB DLM can remove a

wrong letter earlier and assign the right one later as usual. Does not work with GPT partitioned drives.

- New: VolumeSize, VolumeSerial and DeviceType TrueCrypt are determined by means of the partition information instead of the volume information -> prevents delays sometimes
- New: On arrival the volumes are no more accessed unless required to gather information to test for defined criteria; Results sometimes in faster arrivals
- New: UsbDriveInfo command-line params `-rv -rd -rb` to save a report of view Volumes, Drives or BusTypes (upper case R writes Unicode file, lower case Ansi)
- Bugfix: UsbDriveInfo reports was cut at 32KB
- Bugfix: Since V4.7.3 UsbDriveInfo crashed if USB DLM is installed on a non-NTFS formatted drive
- New: UsbDriveInfo can change drive letters when started with admin privileges

V4.8.5.0 (29 January 2014)

- Bugfix: Since V4.8.2 USB DLM Friendly Names did not work as DeviceID criteria
- Bugfix: UsbDriveInfo did not show network drives in the BusTypes view

V4.8.4.0 (28 January 2014)

- Bugfix: Since V4.8.0 USB DLM might crash with `NoMediaNoLetter>0`
- New: UsbDriveInfo with BusTypes view

V4.8.3.0 (23 January 2014)

- Bugfix: Since V4.8.0 USB DLM crashed on arrival of a drive which does not respond within 2 seconds
- Bugfix: Bugfix: Crash UsbDriveInfo if started without a USB DLM.INI
- Bugfix: Command interface returned errors on success

V4.8.2.0 (12 January 2014)

- Bugfix: Since V4.8 crash after many arrivals+removals of a drive
- Bugfix: Using the command interface could cause CPU load and massive logging
- Bugfix: OnServiceStart did not work
- Bugfix: Since V4.8 UsbDriveInfo might show wrong INI sections

V4.8.1.0 (27 December 2013)

- Bugfix: Balloontip for U3 drives often showed only one drive in V4.8.0.0

V4.8.0.0 (25 December 2013)

- Bugfix: Checking drive letters on startup of the USB DLM service did not work for non-USB drives if the drive's BusType was not obvious, e.g. a SATA drive with a DeviceID SCSI...
- Bugfix: AutoRun MD5 check did not work after V4.2.1
- Bugfix: Drives configured in [IgnoreDevices] got the drive letter assigned by Windows removed
- Bugfix: Ignoring the Apple iPod did not work

- New: Events [\[OnDiskArrival\]](#), [OnDiskRemovalRequest], [OnDiskRemoval] for disks, e.g. without partitions or with unknown partitions
- New: [Command interface](#) for restricted users to flush the file cache, dismount, safely remove hardware, etc
- New: UsbDriveInfo output rearranged and extended
- New: UsbDriveInfo shows criteria good for Copy&Paste to the USBDLM.INI (Options -> Show USBDLM criteria)
- New: UsbDriveInfo resolves the container of TrueCrypt volumes and can "eject" them
- Removed: Support of Windows 2000

V4.7.3.2 (09 June 2013)

- Bugfix (again): Crash on some computers

V4.7.3.1 (22 May 2013)

- Bugfix: Crash on some computers

V4.7.3 (20 May 2013)

- Bugfix: Crash if user is member of a huge number of user groups (~ >100)
- New: ListUsbDrives command-line tool replaced by UsbDriveInfo GUI tool; ListUsbDrives is still available [standalone](#)
- New: The criteria UserGroup accepts group SIDs too
- New: Group membership is determined in a different way (now based upon the user's access token); This should work with cached credentials and NetUserGroups=1 is no more required
- New: [DeviceType](#) "WindowsRecovery" for Windows Recovery partitions
- New: USBDLM ensures that USBDLM.EXE and INI are writable to admins only
- New: `_edit-ini.cmd` for opening the USBDLM.INI with admin privileges

V4.7.2 (10 April 2013)

- New: Output of programs started hidden or as "LocalSystem" is redirected to the log-file
- Bugfix: Crash with more than 160 drives of one class
- Bugfix: USB-Portname not determined when the drive is part of a USB composite device
- Bugfix: Crash on removal of a drive if its VolumeName could not be determined on arrival
- Bugfix: Crash UsbDriveInfo if a partition has an alignment >= 4 GB
- Bugfix: When attaching a USB floppy drive a wrong balloontip was shown for a moment

V4.7.1.51 beta (18 September 2012)

- Bugfix: Crash on arrival of a drive who's VolumeName cannot be determined

V4.7.1.50 beta (17 September 2012)

- Bugfix: Crash on removal of a drive who's VolumeName could not be determined on arrival
- Bugfix: Warning on USBDLM.INI in the "Virtual Store" of Vista/Win7/Win8 did not work

- Bugfix: UsbDriveInfo did not show the USBDLM service status without admin privileges

V4.7.1.0 (09 Sept 2012)

- Bugfix: After standby or hibernation drives without a media was removed from USBDLM's internal drivelist; as a result OnRemovalRequest and OnRemoval events did not work anymore
- Bugfix: On Windows 2000 OnArrival and balloon tips did not work with some devices
- Bugfix: No [BalloonTips](#) for TrueCrypt volumes with ShowTrueCrypt=1 and ShowVirtual=0
- Bugfix: AutoRuns sometimes not executed on system start if AutoRunOnLogon=1 and AutoRunOnStartup=0
- New: [Letters by Disk Signature](#)
- New: [StaticLetters](#) - drive letter which USBDLM will never change. This way you can still assign certain drive letters by means of the Windows Disk Management
- New: [DeviceType](#) OtfVolume for virtual drives created by [FreeOTFE](#)
- New: In [OnDeviceArrival](#) sections all criteria not related to a new drive are available
- New: Deleting the [System Volume Information](#) folder works on NTFS drives
- New: Some optimizations

V4.7.0.0 (15 Jan 2012)

- Bugfix: Criteria MaxTimeSinceBoot did not work
- New: [Letters by USB-Port](#) works with non USB drives by generating a port name from the location string
- New: [Letters by Operating System Version](#)
- New: deviceid text files are cached instead of being read each time
- New: UsbDriveInfo shows (when started with -a) active filter drivers for drives and their device classes
- New: UsbDriveInfo shows partition and FAT cluster alignment
- New: Warning balloon-tip when a drive with GUID partition table (GPT) is attached on XP or Windows 2000

V4.6.9.2 (08 Oct 2011)

- Bugfix: Since V4.6 autorun.inf open usually did not work
- New: Timeout for criteria [FileExists](#), [DirExists](#) can be configured
- New: Registry setting for USB [WriteProtect](#) can be set

V4.6.9.1 (11 Sept 2011)

- Bugfix: Since V4.6 the external deviceids.txt file worked as Unicode only
- New: Some [time criteria](#)
- New: Win7 [BusTypes](#) Virtual and VirtualFileBackedVirtual

V4.6.8.3 (10 July 2011)

- Bugfix: Since V4.6 crash on install if WriteLogFile=1 and the log file cannot be created, e.g. due to

missing privileges, invalid path, etc

- Bugfix: Crash when FileExists criteria is evaluated on system and device events

V4.6.8.2 (19 June 2011)

- Bugfix: Since V4.6 Firewire drives had a crippled FriendlyName, e.g. in the BalloonTip

V4.6.8.1 (08 June 2011)

- Bugfix: Some function stopped working after a while in V4.6.8.0

V4.6.8 (05 June 2011)

- Bugfix: Criteria UserGroup did not work on some Vista/Win7 installations since V4.6

V4.6.7 (30 May 2011)

- Bugfix: Passwords did not work under Vista/Win7 since V4.6

V4.6.6 (27 May 2011)

- Bugfix: Section NetworkLetters/ExcludedLetters had no effect since V4.6

V4.6.5 (15 May 2011)

- Bugfix: Since V4.6 possible crash on arrival of multiple network or subst drives at the same time
- New: [Setting](#) VirtualVolumes for deactivating USBDLM dealing with virtual drives (Network, Subst, TrueCrypt and other user specific drives)

V4.6.4 (26 April 2011)

- Bugfix: Some small fixes

V4.6.3 (21 March 2011)

- Bugfix: Setting USB drives to the policy "Optimize for speed" and activating the write cache worked only, if the drive's registry key "Classpn" was already present, which is the case only if the removal policy was manually changed before
- New: New variable %PureVolumeName% gives something like Volume{06588842-39a4-11e0-8027-806d6172696f} which is useful when dealing with tools like [DD](#) which want to see volume names like \\.\Volume{06588842-39a4-11e0-8027-806d6172696f}

V4.6.2 (15 March 2011)

- Bugfix: Since V4.6 settings from the registry did not work correctly

V4.6.1 (23 Feb 2011)

- Bugfix: Since V4.6 no balloon tips under Windows Server 2008 and under Window Vista and Windows 7 if the UAC is deactivated

V4.6.0 (20 Feb 2011)

- New: Changeover to Unicode: USBDLM.INI, BalloonTips, file names etc have no more problems

with codepages and charsets

- New: OnUserActivated and OnUserDeactivated [system events](#)
- New: [Explorer Icon and Label](#) can be defined in the volume's DriveLetters section
- New: Executable files are searched for first in the Windows search path (the PATH environment variable) and then in the working directory
- New: OnArrival for network and subst drives
- New: The FileExists criteria sees user drives (as network, subst and TrueCrypt drives)
- New: The drive letter can be directly extracted from the [Volume Label](#) to the new variable %LetterFromLabel% which can be configured as drive letter
- New: UsbDriveInfo shows started with parameter -v all USBDLM-variables{linkID=}
- New: USBDLM can activate the ~~write cache for USB drives~~{linkID=6}
- New: x64 MSI is available
- New: [USB DLM.WXS](#) file is provided for creating MSI file by means of the [Microsoft WIX](#) toolset, see [Creating an MSI](#){linkID=135}

V4.5.9 (20 Feb 2011)

- BugFix: Since V4.5.1 USBDLM stopped working under very rare conditions which is if all the following is true: USB drive attached which is recognized as cardreader, an [Explorer] section is used, no [OnArrival], no [OnRemovalRequest] and no [OnRemoval] for this drive is configured

V4.5.8 (17 November 2010)

- Bugfix: Since V4.5 USB floppy drives got a high letter assigned instead of keeping them at A: or B:
- Bugfix: Since V4.5.3 USBDLM changed the Windows timer resolution from 10 or 15 ms to 1 ms - the result might be a slightly higher power consumption of the CPU

V4.5.7 (6 November 2010)

- Bugfix: Since V4.5.3 registry settings did not work
- Bugfix: Since V4.5.3 INI settings did not work if the last section name is unknown

V4.5.6 (14 October 2010)

- Bugfix: With five or more drives in the balloon tip the word "drives" was missing in V4.5.5
- New: Multiple DeviceArrival events for the same device are ignored

V4.5.5 (12 October 2010)

- Bugfix: OnDeviceArrival did not work with USBDLM V4.5.4
- New: Some grammar for Slavonic language balloon tips

V4.5.4 (10 September 2010)

- Bugfix: With USBDLM V4.5.3 OnBalloonClick stopped working after standby or hibernation

V4.5.3 (5 September 2010)

- Bugfix: USBDLM prevented roaming user profiles written back to the server on user logoff or

shutdown

- Bugfix: Text_mounted_to_plr for the balloon tips works now
- New: Dock and Undock ~~system events~~{linkID=94}
- New: Handling large USBDLM.INI files accelerated
- New: Some optimization

V4.5.2 (1 August 2010)

- Bugfix: Again problems with Balloontips for devices with quotation marks in the name
- Bugfix: OnArrival sections sometimes executed twice
- Bugfix: Delay on startup of the USBDLM service removed
- Bugfix: USBDLM timeout error in the Windows Event Log on some slowly starting systems
- Bugfix: non responding drives was not tested again, so their OnArrival section was not executed
- Bugfix: BitLocker drives sometimes not detected as such
- Bugfix: Several fixes for Windows 2000
- New: Letters by ~~Computer Name~~{linkID=72}
- New: Letters by running ~~process~~{linkID=73}
- New: UsbDriveInfo shows USB 3.0 Super Speed

V4.5.1 (13 June 2010)

- Bugfix: On resume from standby or hibernation non responding card readers got assigned a drive letter even they should not because of NoMediaNoLetter=1
- Bugfix: When dismounting a TrueCrypt volume in an OnRemovalRequest section then the removal message for the TrueCrypt volume was not processed by USBDLM; in result no OnArrival and autorun.inf was processed on next arrival of this TrueCrypt volume
- Bugfix: The password dialog was not shown on some systems
- Bugfix: UsbDriveInfo tool crashed with latest usb.id file
- New: ~~Password~~{linkID=80} can be received with an anonymous logon

V4.5.0 (9 May 2010)

- New: Multiple AutoRun sections on ~~system events~~{linkID=94}
- New: Vista ~~Scan and Fix~~{linkID=122} can be suppressed
- New: ~~Passwords~~{linkID=80} for drives
- New: ~~AutoRun~~{linkID=90} works for remote users
- New: ~~AutoRun~~{linkID=93} for other devices than drives

V4.4.2 (13 December 2009)

- Bugfix: Variable %UsbSerial% was empty since V4.4.0
- Bugfix: Drive letter swapping on startup not performed when the USBDLM service started after the user's logon
- Bugfix: FileExists when used with wildcards worked only when exactly one file was found

- New: Swapping driveletters on startup can deal with more than two letters

V4.4.1 (11 November 2009)

- Bugfix: Hiding drives from the "~~Safely Remove Hardware~~ {linkID=99}" might not work under Vista, under Window 7 it still does and will not work for "Removable" drives
- Bugfix: UsbDriveInfo always showed a warning about a USBDLM.INI in the Vista/Win7 virtual store
- Bugfix: Delay on service startup on some systems removed
- Bugfix: Close button on the BalloonTip might be drawn incomplete under XP

V4.4.0 (1 November 2009)

- Bugfix: ~~BusTypes~~ {linkID=64} in list notation (e.g. BusTypes=USB,SCSI) didn't work
- Bugfix: Deactivating ~~System Restore~~ {linkID=120} for new drives didn't work
- Bugfix: OnArrival did not work for drives with no media present, even MinDiskSize=- was configured
- Bugfix: UsbDriveInfo now shows all sections effective for a drive
- New: x64 version available
- New: Determining USB port names causes no more delays
- New: Startup of the USBDLM service accelerated
- New: Additional search path for ~~executables~~ {linkID=90}
- New: Alternative notation of the ~~USB Port Name~~ {linkID=44}
- New: Letters by ~~USB-Port~~ {linkID=44} much faster now
- New: Label and Icon from ~~autorun.inf~~ {linkID=90} files can be used
- New: From autorun.inf on x64 systems the section [autorun.amd64] is read first
- New: autorun.inf can be read even redirected by [IniFileMapping](#)
- New: VolumeSerial criteria can be used with wildcards
- New: "~~BitLocker To Go~~" {linkID=97} encrypted drives are recognized as such

V4.3.2 (7 July 2009)

- Bugfix: When only NTFS mountpoints are configured in a section then an assigned drive letter was not removed
- Bugfix: Text color in the balloon tips always black even configured different in Windows
- Bugfix: UsbDriveInfo shows for harddrives and removable drives the correct "Disk DosDevName" (so far it showed the PDO name)
- New: On startup USBDLM can swap drive letters in some situations if required
- New: Section numbers from 1 to 9 can be noted as 01 .. 09, e.g. [DriveLetters01]
- New: Items Letter, Letters, Letter1 bis Letter9 - all understand all formats (single letter, list of letters, mountpoint, INI file)

V4.3.1 (1 June 2009)

- Bugfix: Delay on Windows standby / hibernation removed
- Bugfix: Slow responding drives handled more tolerant
- Bugfix: Firewire drives have again the short, "handmade" name in the balloon tip

V4.3.0 (30 April 2009)

- Bugfix: Volume Label sometimes not read after media change under Vista. The Windows Explorer had the same problem, see <http://support.microsoft.com/kb/943632>
- New: UsbDriveInfo shows the sections from the USBDM.INI which are used for a drive
- New: Faster operation by gathering required drive information only
- New: Multiple NTFS folders {linkID=5} can be assigned as mount points, also in addition to a drive letter
- New: In sections {OnArrival}{linkID=90}, OnRemoval etc up to 10 commands can be executed (open1 to open9 and the known open)
- New: {OnArrival}{linkID=90} and AutoRun.inf for TrueCrypt volumes
- New: When a DeviceID {linkID=39} is configured then the BusType for non USB drives can be omitted, same for the DriveType e.g. for CDROM drives
- New: {OnArrival}{linkID=90} can start a program under Vista with full rights ("elevated")
- New: Criteria UserIsAdmin {linkID=71} is under Vista related to the "elevated" user, so it is 1 for an Admin under UAC
- New: Criteria for absence of a file {linkID=69} (FileNotExists)
- New: {OnRemovalFailed}{linkID=91} is executed asynchronously to avoid event ping pong when using [RemoveDrive](#)
- New: ~~AutoRun on System Events~~ {linkID=94}
- New: Handling of first time arrivals under Windows 2000 optimized
- New: Using {OnArrival}{linkID=90}, OnRemoval etc there is the new default criterion MinDiskSize=1, so a media must be present
- New: Drive letters can be checked ~~on resume~~ {linkID=6} from standby and hibernation mode
- New: Drive letters by ~~Volume Serial Number~~ {linkID=52}
- New: Drive description for USB drives in the menu of "~~Safely Remove Hardware~~ {linkID=99}"
- New: Balloons under Vista in almost correct design
- New: Balloons with fade out effect and shadow (if activated in the Windows settings)
- New: Using USBDM ~~Variables~~ {linkID=} for environment variables can (and should) be configured in detail
- New: Section [ExcludedLetters] renamed to ~~[NetworkLetters]~~ {linkID=14}

V4.2.5 (29th March 2009)

- Bugfix: Crippled BalloonTip if quotation mark in device name
- Bugfix: Sometimes wrong size in BalloonTip and UsbDriveInfo
- Bugfix: The execution of the ~~autorun.inf~~ {linkID=90} open line was in V4.2.4 (again) by default

made with removed admin privileges (AutoRunInfRestricted was on by default)

V4.2.4 (3rd December 2008)

- Bugfix: The execution of `[OnArrival]{linkID=90}` failed when the path contained spaces and command-line parameters are used
- Bugfix: The execution of `[OnArrival]{linkID=90}` failed when the arrived drive has no mountpoint and no WorkDir is specified
- Bugfix: Under Windows 2000 the port names `{linkID=44}` of USB1 devices attached to USB2 ports where not determined

V4.2.3 (14th September 2008)

- Bugfix: The execution of `[OnArrival]{linkID=90}` could be suppressed by holding the Shift key even Force=1 is set
- Bugfix: The execution of the `autorun.inf{linkID=90}` open line was by default made with removed admin privileges (AutoRunInfRestricted was on by default)
- Bugfix: `[NoMediaNoLetterInclude]{linkID=29}` didn't work in most cases

V4.2.2 (17th August 2008)

- Bugfix: `[OnArrival]{linkID=90}`, `[OnRemoval]{linkID=91}` works again without configuring a workdir when the drive has no mountpoint
- Bugfix: `[OnRemoval]{linkID=91}` works again without configuring a workdir
- Bugfix: `[OnRemoval]{linkID=91}` is triggered on removal of a media from a card reader
- Bugfix: New `BusType{linkID=64}`s SD and MMC
- Bugfix: Virtual drives of the Vista's "Volume Shadow Copy" will not get a drive letter assigned
- Bugfix: Delay on Windows shutdown removed

V4.2.1 (19th April 2008)

- Bugfix: On `autorun.inf`, `[OnArrival] etc {linkID=90}` the root of the drive is the working directory again

V4.2.0 (12th April 2008)

- Bugfix: problems with NTFS mountpoint longer than 51 chars fixed
- Bugfix: AutoRun programs did not become the active window under Vista
- New/Bugfix: unavailable network drives considered
- New: `NoMediaNoLetter{linkID=29}` works under Windows 2000
- New: most settings take effect without restarting the USB DLM service
- New: `Wildcards{linkID=115}` can be used configuring several parameters
- New: drives can be completely ignored `{linkID=105}`
- New: drives and other devices can be removed from or added to the `safely remove hardware{linkID=99}` facility
- New: the policy "Optimized for performance" `{linkID=6}` can be set for USB drives

- New: on ~~AutoRun~~[{linkID=90}](#) USBDLM variables can be made available to the executed program as environment variables
- New: on ~~AutoRun~~[{linkID=90}](#) programs can be executed without admin privileges (if the current user is an admin)
- New: obsolete NTFS mountpoints can be removed on startup

V4.1.0 (19th October 2007)

- Bugfix: Last line in the USBDLM.INI had been ignored when fewer than two empty line at the end
- Bugfix: Even with NoMediaNoLetter=0 USBDLM registered for some notifications. Under Vista64 this prevented drives from being 'prepared for safe removal'
- Bugfix: OnRemoval works now on removal of a card from a card reader
- Bugfix: USBDLM service no more marked as 'interactive' under Vista which caused warnings in the syslog
- Workaround: ~~NoMediaNoLetter~~[{linkID=29}](#) works under Vista64
- New: ~~NoMediaNoLetter~~[{linkID=29}](#) can be configured more detailed
- New: ~~NoMediaNoLetter~~[{linkID=29}](#) can be forced for certain drives by a section [NoMediaNoLetterInclude] (analog to [NoMediaNoLetterExclude])
- New: Drive letters in ~~short notation~~[{linkID=4}](#), like Letters=R,X,Y,Z
- New: User=- as criterion for 'no user logged on'

V4.0.2 (21th August 2007)

- Bugfix: UsbDriveInfo output had Unix style end of line marks (CR only) instead of DOS style (CF+LF) and Notepad cannot show this
- Bugfix: AutoRun events for drives without a drive letter did work only when an additional criterion Letter=- was configured
- Bugfix: Under Windows 2000 partition numbers and balloon tips didn't work correctly for hard drives with multiple partitions

V4.0.1 (20th July 2007)

- Bugfix: ExcludedLetter didn't work
- Bugfix: ListUsbDrives required admin privileges to show all information

V4.0.0 (15th July 2007)

- New: new, more simple, but incompatible format for the USBDLM.INI
- New: drive letters depending on the logged on user's name, group or its admin status
- New: drive letters depending on some new criteria
- New: AutoRuns on media instertion (CD/DVD too)
- New: AutoRuns depending on several criteria (as volume label, drive letter, device id and others)
- New: AutoRuns on removal request and after removal of a drive
- New: BalloonTip texts in several languages

- New: Runs on balloon click
- New: Explorer default icon and label depending criteria
- New: Drive events into the Windows system log

V3.4.2 (9th May 2007)

- Bugfix: multiple wrong drive letters shown in balloon tip when hardware detection incomplete
- Bugfix: detection of U3 drives improved

V3.4.1 (1st May 2007)

- Bugfix: V3.4.0 bases upon an old codebase - please update to this version. All the new features promised for V3.4.0 now really work.

- Bugfix: open= in autorun.inf works too if the file to execute is located on the attached drive

V3.4.0 (18th April 2007)

- New: drive letters by size can use the size of the disk instead of the partition
- New: drive letter depending on the volume label ("drive name")
- New: USBDLM.INI on the attached drive for non USB drives too
- New: device IDs for [DriveLettersDeviceIDx] optionally from external file
- New: settings optionally in the registry instead the INI

V3.3.1 (20th March 2007)

- Bugfix: Possible crash on machines more that one CPU core or with HyperThreading
- Bugfix: Errorlevel on installation with -silentinstall was still swapped - now 0 on success
- New/Fix: Removal of drive letters for NoMediaNoLetter is done with some delay to prevent rare problems with Win Explorer
- New/Fix: Shell notification about assigned drive letter done with some delay to fix problems with Win Explorer's folder tree view

V3.3.0 (8th Feb. 2007)

- Bugfix: Errorlevel on installation was swapped - now 0 on success
- New: Problem with AutoRun under Vista and Windows 2000 fixed
- New: Installation, deinstallation, start und stop request admin privileges if required
- New: ~~AutoRun~~ {linkID=4} can be secured by a key
- New: non USB card readers stay untouched - too much trouble
- New: no assignment of a new drive letter for drives/partitions without a drive letter when it is a TrueCrypt container
- New: Lots of Finetuning :-)
- New: instead of the approved HLP format the help is provided as CHM because Vista doesn't support HLP anymore

V3.2.0 (11th Dec. 2006)

- Bugfix: DeleteMountpointOnRemoval didn't work since V3.0
- New: 'no letter' can be configured to get rid of certain drives or to prevent users from using them
- New: drive letters can be configured by drive's device ID sting
- New: balloon tips can be disabled for certain drives

Note: This version had a wrong version number '3.1.6.1' until 6th Jan. 2006.

V3.1.0 (6th Nov. 2006)

- Bugfix: Balloontips didn't work for the second user when Fast User Switching is used under XP and always under Vista
- Bugfix: Autorun function started programs in the context 'LocalSystem' instead of the current user's context
- New: drive letters for USB floppy drives

V3.0.3 (22th Oct. 2006)

- Bugfix: USBDLM V3.0.2 crashed with NoMediaNoLetter=1
- Bugfix: detection of conflicts with network shares and subst drives didn't work sometimes
- New: detects card readers with two slots as multis-slot cardreader if they have 'reader' in their device name
- New: should detect IDE and PCI card readers

V3.0.2 (16th Oct. 2006)

- Bugfix: USB floppy drives lost their drive letter when attached without inserted media

V3.0.1 (7th Oct. 2006)

- Executing AutoRun and RunOnArrival the new drive is set as current folder so programs or batch files are found there

V3.0.0 (3rd Oct. 2006)

- All drive letter settings [{linkID=6}](#) are applied now on startup too
- new funktion of section [\[ExcludedLetters\]{linkID=14}](#): These letter will never be used anymore for USB drives
- Read additional [\[ExcludedLetters\]](#) from a central USBDLM.INI
- detects and handles [U3 autorun drives{linkID=54}](#)
- Execution of the open= line in an autorun.inf on removable drives
- Global AutoRun on arrival of any USB or Firewire drive
- ~~Balloon-Tips~~ [{linkID=6}](#) with information about the assigned letters

V2.4.4 (7th Sept. 2006)

- Adaption: USB devices with two drives are no longer seen as multi-slot cardreader. The limit is three drives now because there are USB pen drives with two independent drives

V2.4.3 (28th August 2006)

- Bugfix: Under Windows 2000 you got 'insert disk' error messages when a USB drive without media is attached
- New/Fix: service name changed from 'USB Drive Letter Manager' to 'USB DLM'

V2.4.2 (16th June.2006)

- Bugfix: V2.4.1 didn't start on some machines when booting
- can write a logfile for debugging
- can act as 'eject server' for the command-line tool EjectMedia

V2.4.1 (11th June 2006)

- USB floppy drives are ignored
- problem with double created mount points for multi-slot card readers fixed
- misspelled service name corrected ('USB drive letter manager')
- corrected errorlevels after -install and -silentinstall

V2.4 (21st May 2006)

- can detect drives which got no mount point at all and assigns the first free letter (if nothing else is configured)
- can delete mount points (NTFS folders) when the mounted drive is removed
- mounting to NTFS folders by device name or disk name
- drive letters for USB CD and DVD drives

V2.3 (5th May 2006)

- New: can detect network shares of the currently logged on user, so when Windows assigns the letter of a network share USB DLM remounts it to the next letter that is really available, even nothing is configured

V2.2.2 (18th April 2006)

- Improvement: Works with drives too that need more than 10 seconds between attachment and appearance of the drive letter

V2.2.1 (2nd April 2006)

- Bugfix: works again under Windows 2000 (V2.2 didn't)

V2.2 (30th March 2006)

- works with Windows Vista
- scan for c't USB Agent deactivated by default

V2.1 (19th March 2006)

- drive letter assignment depending on the USB port for USB drives too that appear as 'Local drive'
- remount of multislot cardreader's slots to letters too

V2.0 (14th March 2006)

- drive letter assignment depending on the USB port (for removable drives only)
- remount of multi-slot card reader's slots to an NTFS folder
- using empty NTFS folders as mount points instead of drive letters (experimental)

V1.1 (15th Jan. 2006)

- Some timeouts made longer
- Detection of USB and Firewire works under Windows 2000 too

V1.1.1 (17th Jan. 2006)

- Stupid bug in the detection of USB and Firewire fixed that caused an access violation

V1.0 (10th Jan. 2006)

- Allocation of drive letters depending on the size of the attached drive
- Exclude list for drive letters to overrule a USB DLM.INI on the newly attached drive
- Exclude list for volume labels
- No remounting when a USB Agent.inf is on the drive and the c't USB Agent is running
- Freeware for private and nonprofit educational use only

V0.6 (15th Dec. 2005)

- Check for a media in drive not longer done thru its volume serial number because drives often get no number by the vendor

V0.5 (24th Oct. 2005)

- Handles drives only that are removable or hotplug devices

V0.4 (5th Sept. 2005)

- Handles fixed drives too because there are USB flash drives with a drive type 'DRIVE_FIXED' instead of the usual 'DRIVE_REMOVABLE'

V0.3 (25th Aug. 2005)

- First public version

Software licence USBDLM

This licence agreement is based on and bound to German law. This is a translation of the most important facts for our customers and visitors. All programs and documents were created corresponding German law. If there were any infringements of changed or international law it is done unintentional. Please let me know in order to allow corrections.

1. The subject of the contract

The subject of the contract is the enclosed software, the software description and instructions, as well as other associated written material, in the following also called 'software'.

The author makes attentive to the fact that it is not possible to provide software that works error free in all applications and combinations.

The subject of the contract is therefore only a software, which is usable in the sense of the program description.

2. Granting of a licence

The author grants public schools, universities and other non profit institution of education where the students outnumber all others a free licence.

Furthermore public libraries whose usage is free of charge can use the software for free.

Other users have 30 days to determine if this product meets their needs. After this time one licence per computer must be ordered or the software must be removed from the computers.

The licence applies to all version numbers of the software now and in the future to the appropriate number of licences.

The volume licences are valid for the particular number of computers within a company or organization.

The author does not supply updates since all updates are offered for download.

The software can be copied and passed on freely, as long as the ZIP or MSI archive is unchanged. Re-packaging is allowed for internal use only.

3. No further legal claims

The author is both author and owner of the software, as well as of algorithms and procedures used in this software.

4. No modifications or decompiling

Changes in any files, disassembling, reverse engineering, patching of this software is expressly prohibited.

5. Redistribution

Redistribution of the unmodified software archive files on CD/DVD media of computer magazines is allowed.

Making them available for download is unwanted. Whenever possible link to <http://www.uwe-sieber.de/files/usbdm.zip> and <http://www.uwe-sieber.de/files/usbdm.msi>

Re-packaging is allowed for internal use only.

Offering the software for download by means of a "downloader" software is not allowed.

6. No warranty

I make no guarantee, implied or otherwise as to the accuracy of the documentation or the software's suitability for any purpose. It is a condition of the licence that the user accepts that I'm not responsible for any damages arising by the use or misuse of this software. This includes loss of profit, data, any material losses and their consequential damages: even if I'm informed about the possibility of those damages. I try to fix bugs as soon as possible.

Installation

Unzip the distribution file (USB DLM.ZIP), e.g. to your programs folder (a USB DLM folder is included).
On NTFS formatted drives this requires admin privileges.
As result the files should be in a folder like C:\Program Files\USB DLM

Under Vista and Windows 7/8 I suggest to install USB DLM in a different folder than C:\Program Files, because under this folder changes of the USB DLM.INI made by a non elevated user are redirected into the "Vista Virtual Store" where USB DLM will not read it.
In a folder like C:\Tools\USB DLM there is no problem. Or just always open the INI file by means of the _edit-ini.cmd, it asks for admin privileges if required.

USB DLM will not run from a network or subst drive, it must be located on a "real" local drive.

Then enter your preferred configuration into the USB DLM.INI (see below). There is a sample file included called USB DLM_sample.INI. Either rename the sample file or just run the _edit-ini.cmd, it creates and opens a basic USB DLM.INI if there is none.

On an NTFS formatted drive editing the USB DLM.INI requires admin privileges. Under Vista/Win7/8 you should start the Editor elevated, otherwise the USB DLM.INI might be redirected into the Vista virtual store where it has no effect. This is what the _edit-ini.cmd does.

If you are logged on with administrator privileges you can simply register the USB DLM service.

```
_service_register.cmd
```

This does not copy any files, it only registers USB DLM as service and starts the service.
Without administrator privileges you will be asked for.

Once installed USB DLM is permanently running as Windows service. The service starts automatically on Windows startup.

Remember: Once installed as service it does not matter from where the service is started. The USB DLM.EXE is loaded from the folder where the _service_register.cmd had been started from!

Deinstallation:

You can deregister the service using parameter -deregister, this is what the _service_deregister.cmd does:

```
_service_deregister.cmd
```

This stops the service and deregisters it. It does not remove any files.

Silent

Installation and deinstallation can be done without the message box by using

USB DLM -silentregister
USB DLM -silentderegister

It returns Errorlevel 0 on success, 1 on failure.

Start and stop the USB DLM service

You can stop and start the installed service by means of the NET command:

```
net stop USB DLM  
net start USB DLM
```

or by means of the `_service_stop` and `_service_start` command script which are using the USB DLM.EXE and request admin privileges if required.

You can change the drive letter settings in the USB DLM.INI without restarting the service because it's checked for changes each time a drive event occurs.

Since V4.2 most other settings take effect without restarting the service. The only exception are settings which result in registration for device notifications (as `NoMediaNoLetter=1` or `[OnRemovalRequest]`). This is done then when the device in question is reattached or the service is restarted.

See [configuration](#).

Hint:

The command scripts contain simple command lines like
`"%~dp0USB DLM" -register`

`%~dp0` is the placeholder for the **d**rive and **p**ath to the CMD file. How does it work? `%0` is placeholder for the running CMD file like `C:\Program Files\USB DLM_service_register.cmd`.
`~dp` is a "modifier" which delivers drive and path like `C:\Program Files\USB DLM`. Because of the possible space character in the path it must be quoted.

Read more about batch parameters and modifiers here:

<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/percent.mspx>

Configuration

The desired drive letters or mount points and other settings are defined in a text file called USBDLM.INI located in the same place as the USBDLM.EXE.

Modern applications often store their settings in the Windows registry but I don't like that. INI files are the 'classic' approach.

Windows Vista, Windows 7, 8, 8.1, 10, Server 2008 (R2), 2012:

Since Windows Vista the "Virtual Store" might redirect the INI file to C:\Users\<(UserName)\AppData\Local\VirtualStore\Program Files\USB DLM when you modify or copy it.

But the USB DLM service always reads from its own folder. So, since Vista always edit the INI file as "real" Administrator, this is what the _edit-ini.cmd file performs.

Or install USB DLM into a different folder, for instance to C:\Tools\USB DLM because the Vista Virtual Store deals with the "C:\Program Files" folder only.

I suggest to open the INI by means of the _edit-ini.cmd because it requests the required admin privileges.

_edit-ini.cmd also creates a small USB DLM.INI if there is none present yet.

Unicode:

If you need Unicode then save the INI file as such, it must be Little Endian, with byte order mark FFFE, the Windows Editor (notepad) can do correctly.

How INI files work:

INI files have sections leaded in by names in square brackets, e.g. [DriveLetters] or [BalloonTips]. The settings relevant to that section appear below the section name.

Comment lines begin with a semicolon. Never try to deactivate a section this way:

```
:[DriveLetters]  
Letters=X,Y,Z
```

This extends the previous section to the begin of the next section which leads to unexpected results. Either comment-out all lines of the section or just invalidate the section name like so:

```
[xxxx DriveLetters]
```

It is still a new section but with no effect because of the unknown name.

In regular INI files each section name must be unique, same for values within a section. With USB DLM V5.0 this rule is no more. You can have the same section name many times, the priority of unnumbered sections is defined by order of appearance.

But this is the case only for sections with criterions as DriveLetters, OnArrival, OnBalloonClick etc, but not

for Settings, BalloonTips etc.

Settings in the Registry:

Since V3.3.1 USBDLM can read its settings from the registry too. It reads from HKLM/Software/Uwe Sieber/USBDLM

If this registry key exists, then the USBDLM.INI is ignored! Only the log file settings are read from the INI then.

To make USBDLM see changes in the registry, change the (Standard) value under HKLM/Software/Uwe Sieber/USBDLM to any different value than before. USBDLM reads the configuration again then.

Here is a sample REG file (do not execute it unchanged): http://www.uwe-sieber.de/files/usbdm_sample.reg

Under x64 Windows the Win32 edition of USBDLM is redirected to HKLM/Software/Wow6432Node/Uwe Sieber/USBDLM so better use the x64 version here.

Simple Samples

Without a configuration USBDLM solves conflicts between USB drives and network or subst drives.

USBDLM offers several ways to control the drive letter assignments. For each definition there is a section in the USBDLM.INI with up to 9 drive letter lines.

The value numbers have no relation to anything. The values are just evaluated from low numbers upwards to high numbers.

Most simple sample that works for USB flash drives and USB harddrives:

Before USBDLM V5.0 with numbered values:

```
[DriveLetters]
Letter1=U
Letter2=V
Letter3=W
.
.
Letter9=X
```

Since USBDLM V5.0 without numbers:

```
[DriveLetters]
Letter=U
Letter=V
Letter=W
.
```

.
Letter=X

Each letter is checked then and the drive will be remounted to the first one that is available.

Since V4.1 you can use a **short notation**. This works for drive letters only, not for NTFS mount points!
Samples:

```
[DriveLetters]  
Letters=A,W,X,Y,Z
```

You can use ranges too:

```
; W-Z is expanded to W,X,Y,Z  
[DriveLetters]  
Letters=A,W-Z
```

```
; Z-W is expanded to Z,Y,X,W  
[DriveLetters]  
Letters=A,Z-W
```

And you can omit the comma:

```
[DriveLetters]  
Letters=AWXYZ
```

Since V4.3.2 Letter, Letters, Letter1 to Letter9 are all handled equal, so you can even write
Letter1=AWXYZ.

By default USB DLM assigns one mountpoint only. Since V4.3 multiple [NTFS mountpoints](#) can be assigned, also in addition to a drive letter.

Sample for letter U: **or** V: and two additional NTFS folders:

```
[DriveLetters]  
MaxMountPoints=3  
Letter1=U  
Letter2=V  
Letter3=C:\_USB\%DevName%  
Letter4=C:\_Share\Drive_%DriveLetter%
```

Windows does not allow to assign more than one letter as mountpoint.

Priority

First "Letters" is read, then "Letter" and Letter1 to Letter9. Since V5.0 then up to 9 additional Letter lines.

No letter / remove the drive

To hide a drive (remove it's drive letter) configure a single minus char:

```
Letter=-
```

To remove a drive (prepare it for safe removal) configure two minus chars:

```
Letter=-
```

To get partition number 1 on a USB disk at X: and hide partition 2:

```
[DriveLetters]
```

```
Letter=X
```

```
Letter=-
```

But it works only if X: is available. Otherwise all partitions get no letter.

To hide partition number 1 on a USB disk and get partition 2 at X:, this will **not** work:

```
[DriveLetters]
```

```
Letter=-
```

```
Letter=X
```

Why? For each drive the configured letters are checked and the first available one is used. "No letter" is always available, so the second Letter line is never reached. The drive letter of each partition is removed. Use [Drive Letters by Partition Number](#) in this case.

Drive letters depending on certain criteria

To apply a section to certain drives only, you can define criteria in this section. If you use multiple then since V5.0 there is no more need for numbering. It still works but is no more required.

The sections are tested from [DriveLetters1] to [DriveLetters99] and finally 100 sections [DriveLetters] (without a number) in order of appearance.

So, configure special cases first and more general ones towards the end.

USB DLM uses the drive letters from the first section that fits only. There is no fallback to another section if all configured drive letters in a section are in use!

Sample to lock drives above 10 GB and get smaller drives at U, V and W:

```
;remove the drive letter for drives larger than 10GB
```

[DriveLetters]
MinDriveSize=10GB
Letter=-

;all others at U, V, W

[DriveLetters]
Letter=U
Letter=V
Letter=W

or in short notation:

[DriveLetters]
Letters=U,V,W

There is a detailed description under [Drive letters depending on certain criterions](#).

[General Settings](#)

Mounting to NTFS folders

Instead or in addition to letters you can set up empty folders on NTFS drives as mount points too, see [Mounting to NTFS folders](#)

Mounting to NTFS folders

Instead of letters you can set up empty folders on NTFS drives as mount points too.
If the target folder doesn't exist, then USBDLM creates it.
Once created the folder can be renamed. USBDLM accepts this as long as it is in the same base folder.

When using NTFS folders as mount point, you can use the drive's device name or disk name for the folder name:

Configure '%DevName%' or '%VolumeLabel%' which USBDLM replaces with the drive's device name or the volume label ('disk name') resp.

Sample: Assumed the drive's Friendly Name is 'Corsair Flash Voyager'.

```
[DriveLetters]
```

```
Letter=C:\_USB\%DevName%
```

Then USBDLM would create "C:_USB\Corsair Flash Voyager" and remount the drive to it.

'%DevName%' and '%VolumeLabel%' are some of the USBDLM [variables](#) {linkID=}

In contrast to drive letters, folders used as mount points are not removed when the mounted drive is removed. But USBDLM removes the folder.

If it not shall do so:

```
[Settings]
```

```
DeleteMountPointsOnRemoval=0
```

If the drive is removed while the USBDLM service is not running, e.g. after Windows has been shutting down, then the removal of the folder fails of course. This can be done on next startup or when a user logs on. Configure the folders which contains the NTFS mountpoints.

Sample matching to the DriveLetters section sample above:

```
[DeleteUnusedNtfsMountpoints]
```

```
Folder=C:\_USB
```

```
Folder=
```

Since V4.3 USBDLM can assign multiple NTFS folders; also in addition to a drive letter. For compatibility this is disabled by default.

Sample to get a drive at U: if available (the first free letter otherwise) and two NTFS folders:

```
[DriveLetters]
```

```
MaxMountPoints=3
```

```
Letter=U
```

```
Letter=C:\_USB\%DevName%
```

```
Letter=C:\_Share\Drive_%DriveLetter%
```

Sample to get a drive at U: if available and no letter otherwise and two NTFS folders:

```
[DriveLetters]
MaxMountPoints=3
Letter=U,-
Letter=C:\_USB\%DevName%
Letter=C:\_Share\Drive_%DriveLetter%
```

The default value for MaxMountPoints is 1 and can be set in section Settings to change it globally.

```
[Settings]
MaxMountPoints=3
```

Mounting into NTFS folders brings some **drawbacks**: Several drive specific functions are no longer available in the Windows Explorer like autorun, eject, defrag, format...

If the drive letters appear insistently again, then another software is working which checks the drive letters and "repairs" them.

The "U3 launchpad" of U3 flash drives is known for doing so when it's started (the U3launch.exe on the fake CDROM drive).

The Windows Explorer sees the mounted drive and the host drive as the same. Therefore drag'n drop does move files instead to copy them.

When a file is deleted in the Windows Explorer then it is moved by default to the Recycle Bin which is a folder in the root of the drive. Only the reference to the file is moved, the contents stay untouched. But when a file is deleted that is located on a drive that is mounted into an NTFS folder, the moving doesn't work, the file must be copied then which may take a while. Deleting a folder to the Recycle Bin doesn't work at all, XP shows a misleading error message. Since Vista it works correctly.

So, under XP hold down the Shift key when you delete a file to skip the recycle bin or deactivate the recycle bin for the host drive.

In the "Safely remove hardware" dialog NTFS mount points are unsupported up to including Windows Vista, so you have no idea which drive is which there (unless ~~USB DLM makes readable names there~~ [{linkID=99}](#))

To get rid of cardreader's drive letters, use USB DLM's ~~No Media No Letter~~ [{linkID=29}](#) feature.

Problem: Windows Portable Device driver does not start

Under Vista, Windows 7 and surprisingly Server 2008 for each USB drive Windows also installs a Windows Portable Device (WPD) driver. But the WPD driver relies on a drive letter for no reason. For drives without a drive letter their WPD driver fails with Code 10 (device cannot start). Ask Microsoft why.

This is not a problem with USB DLM, it is just because a drive having no drive letter assigned.

This is fixed since Windows 8.

Settings

Even the section [Settings] is mentioned here again and again, all settings have to be together within the one and only section [Settings]! There can be only one...

Most settings take effect immediately.

Check drive letters on startup

All settings are applied on startup too to catch internal card readers and drives that are attached while booting.

If USBDLM shall not do so:

[Settings]

CheckLettersOnStartup=0

Upcoming network drives cannot be considered here because they do not exist at this time and USBDLM cannot look into the future. Therefore network drives that shall be considered must be configured in section [\[NetworkLetters\]](#).

Check drive letters on user logon / user switch

Under XP and higher all settings are applied on user logon to take user depending settings effect when a different user logs on.

If USBDLM shall not do so:

[Settings]

CheckLettersOnLogon=0

Windows 2000 does not support the required notification messages, so USBDLM cannot check drive letters on user switches.

Check drive letters on resume from standby or hibernation

Some Windows installations wake up with screwed up drive letters after resume. If required, USBDLM can check the letters on resume. This is deactivated by default because it's usually not required.

[Settings]

CheckLettersOnResume=1

AutoRuns on user logon / user switch

Configured AutoRun events are executed by default on arrival of a drive or on insertion of a media only. If USBDLM shall do this on user logon or user switch for drives which are already present:

```
[Settings]
AutoRunOnLogon=1
```

With an auto logon the USBDLM service may not be started yet on logon, so it misses the logon event. Workaround: When there are fewer than two minutes since the system is started and the a user is already logged on when the USBDLM services starts, then the AutoRuns are executed too.

Windows 2000 does not support the required notification messages, so USBDLM cannot execute AutoRun on logon.

Safe command lines

On command lines which contain spaces USBDLM tries to guess where the executable ends and the params begin, e.g.

```
open="C:\Program Files\tool.exe"
```

could be either C:\Program Files\tool.exe as executable without params or it could be C:\Program.exe with Files\tool.exe as param.

USBDLM would execute C:\Program Files\tool.exe but there are more difficult cases where it might execute the wrong one.

With SafeCommandLines=1 USBDLM does not accept command-lines which contain spaces but the executable file not quoted.

Samples of invalid command lines:

```
open=C:\Program Files\tool.exe
```

```
open=C:\Tools\tool.exe /x
```

Samples of valid command lines:

```
open="C:\Program Files\tool.exe"
```

```
open="C:\Tools\tool.exe" /x
```

Default is 0 to prevent existing configurations failing but the USBDLM_sample.ini has it set 1:

```
[Settings]
SafeCommandLines=1
```

UsbDriveInfo scans the settings for ambiguous / invalid command-lines and shows are warning or error depending on the SafeCommandLines setting.

Delete NTFS folders on drive removal

In contrast to drive letters, folders used as mount points are not removed when the mounted drive is removed by Windows. USBDLM removes the folder by default.

If it no shall do so:

[Settings]

DeleteMountPointsOnRemoval=0

Policy "Optimize for performance" for USB drives

Under XP and higher for USB drives there are the policies "Optimize for quick removal" (default) and "Optimize for performance". For the latter Windows pretends to activate a write cache. But it does not when the drive is a "Removable drive" as most of the USB flash drives are.

The only effect is that it's allowed then to format them with the NTFS file system. And with NTFS, XP and higher indeed activate a write cache, also on other computers because the write cache on NTFS drives is independent of the policy. It's all about the permission to format a "Removable drive" with NTFS.

For FAT formatted USB harddrives "Optimize for performance" indeed activates a write cache. With NTFS there is always a write cache.

USB DLM can set the policy "Optimize for performance" for all USB drives:

[Settings]

OptimizeUsbDrivesForSpeed=1

For drives attached for the first time it works immediately. Existing drives must be reattached two times (on first reattach the entry is made, on second reattach it takes effect).

The Windows 10 Task Manager shows USB removable drives on its Performance tab when they are "optimized for performance".

Force the activation of a write cache USB drives

Under XP and higher FAT formatted USB drives have no write cache by default. For USB harddrives (DriveType FIXED) a write cache can be activated by setting the policy "Optimize for performance" while this has no effect for removable USB drives (DriveType REMOVABLE).

By a registry value the activation can be forced, its name is WriteCacheEnableOverride, setting it to 1 activates a write cache even on FAT formatted USB removable drives.

Of course it is required then to use the "Safely remove hardware" facility or another tool as my [RemoveDrive](#) or [EjectMedia](#) to prevent data losses.

USB DLM can set the value to 1 for all USB drives:

[Settings]

UsbWriteCacheEnable=1

For drives attached for the first time it works immediately. Existing drives must be reattached two times (on first reattach the entry is made, on second reattach it takes effect).

- UsbWriteCacheEnable=0 -> set WriteCacheEnableOverride to 0
- UsbWriteCacheEnable=1 -> set WriteCacheEnableOverride to 1
- UsbWriteCacheEnable=-1 -> do not change WriteCacheEnableOverride (default)

External mount events

When drive letters are changed by another software then a service as USBDLM isn't notified off hand. To get notified a simple solution is an invisible Window. Since Vista and under XP when more than one user is logged on a window is not good enough, a window in the context of the interactive user is required. That's what the USBDLM_usr.exe is for. It receives the notifications and sends them down to the USBDLM service.

If this is all not required:

```
[Settings]
ExternalMountEvents=0
```

The OnArrival events and the balloon tips for mounted TrueCrypt, Network and Subst drives depend on this too.

By default USBDLM determines if the USBDLM_usr.exe is required by looking at [Settings] ExternalMountEvents, [Balloons] ShowVirtual and the used OnArrival/OnRemoval sections.

Loading the USBDLM_usr.exe can be controlled manually, see [Technical Information](#) too:

```
[Settings]
LoadUsbdm_Usr=0
```

- 0: never load
- 1: load as required (default)
- 2: always load (with UAC for the restricted user)
- 3: always load (with UAC for the restricted and for the elevated user)

Volume Information Timeout

When looking for a file and when determining the volume label, volume serial and file system name USBDLM waits up to this time in Milliseconds before it gives up. Default is 10 Seconds:

```
[Settings]
VolumeInformationMaxWait=10000
```

Start the Windows Disk Management for uninitialized disks

When a drive arrives which has only zeros in the first sector then USBDLM can open the Windows Disk Management which then asks to initialize it.

```
[Settings]
StartDiskMgmtForUninitializedDrives=1
```

- 0: never
- 1: for USB harddrives (default)
- 2: for all harddrives

Debug Information

If there are problems I may figure out what is wrong by reading the debug logs which USBDLM writes. It can write a log file or live output that can be grabbed by tools as [SysInternals DebugView](#) or [TraceSpy](#).

To activate the log file set WriteLogFile=1, for the live output WriteDebugOutput=1.

To control how detailed the output is, set the LogLevel=1 to 5, 3 is usually good enough.

The file is _USB DLM.LOG in the root folder of the Windows drive by default but can be changed:

[Settings]

WriteDebugOutput=1

LogLevel=3

WriteLogFile=1

LogFile=C:\USB DLM.LOG

Since V4.2 no restart is required to take effect. The log file settings are read from the INI only, even when the Registry is used for other settings.

On startup of the USB DLM service the USB DLM.LOG is deleted. To avoid losing interesting logs USB DLM creates two backups by default, so USB DLM_BAK2.LOG is deleted, USB DLM_BAK1.LOG renamed to USB DLM_BAK2.LOG and USB DLM.LOG is renamed to USB DLM_BAK1.LOG.

The number of backup files can be set:

LogFileBakCount=2

If you need the date in front of each log line:

LogShowDate=1

If you don't need Timing, Loglevel and Thread-ID:

LogShowTech=0

Flush and Dismount on Standby and Hibernation

When Windows enters Standby or Hibernation mode USB DLM flushes the file cache of all volumes. But the file systems stay mounted, therefore it is not a good idea to boot another operating system and mount the file system here too. This will lead to inconsistencies.

USB DLM can try to dismount all volumes on standby or hibernation:

[Settings]

DismountOnSleep=1

Dismounting succeeds only if there are no open files on a volume. The dismounting can be forced for USB and other hotplug drives by setting DismountOnSleep=2 but this makes all open file handles invalid, so on resume applications might complain or fail.

DismountOnSleep=3 forces it for all types of volumes.
Since V5.1 sleeping drive are not woken up for this unless DismountOnSleep=4..

Of course the system volume is not dismounted.

Removed write access on USBDLM.EXE and USBDLM.INI

By default USBDLM checks on startup if USBDLM.EXE and USBDLM.INI can be written by non-admins. If so, then it changes the file privileges. If you don't want this:

```
[Settings]
FixUsbdmFilePrivileges=0
```

Command interface for programs with limited privileges

Some useful Windows function require admin privileges even they are harmless, e.g. flushing a drive's write cache.

Since V4.8 USBDLM has a command interface for some functions. Which functions are available can be configured:

Values of the function, add them if more than one is desired:

no function	0 (default)
flush the write cache	1
dismount a volume	2
eject a media	4
force dismount/eject	8
safely remove hardware	16
take volume on/offline	32
reactivate code 21/47 dev	64

Sample for all but the "force" one: (1+2+4+16+32+64):

```
[Settings]
CommandInterface=119
```

Sample for all: (1+2+4+8+16+32+64):

```
[Settings]
CommandInterface=127
```

Of course you can just set all bits by means of 255, 65535, 0xFF etc.

A USBDLM restart required to take effect.

Locking a volume is the prerequisite for dismounting correctly because it ensures that no one else is accessing the volume. Dismounting without a lock means dismounting by force. Open files stay open but accessing them leads to an error. The affected software will not like this. But if the open files are no more

accessed anyway then there is no problem.

Tools which use the functions:

- [FFB](#): Flushing the write cache and dismount
- [EjectMedia](#): Flushing the write cache, dismount, eject, take volume offline
- [LoadMedia](#): Loading a media, take a volume online
- [RemoveDrive](#): Flushing the write cache, dismount, eject, reactivatable safely remove hardware
- [RestartSrDev](#): Reactivating "Code 21/47" devices

Network Letters

Letters configured in section [NetworkLetters] are not accepted being used for local drives.

They overrule letters configured in external INI files too.

Here you should enter drive letters used as network shares or as subst drives. This way the USBDLM service can consider them even they are not created yet, e.g. at boot or logon time. Furthermore drives which are created later, as TrueCrypt drives, should be configured here.

Up to V4.2 the name of this section was [ExcludedLetters], which still works for compatibility. It was renamed because many users configured the letters of their internal drives here which is wrong and dangerous:

Do not configure letters of local drives here, USBDLM might remount them because they are then "reserved" for network/subst/truecrypt use!

Alternative: Just configure enough drive letters in the DriveLetters sections.

In contrast to other config sections here are Letters1 to 26 read because some users reported that 9 letters are not enough.

Sample to exclude F and G:

```
[NetworkLetters]
Letter1=F
Letter2=G
Letter3=H
```

or short:

```
[NetworkLetters]
Letters=F,G,H
```

or using a range:

```
[NetworkLetters]
Letters=F-H
```

In this section only letters are allowed.

If you want to admin the settings from a central point, the USBDLM.INI can be read from a network share:

[Settings]

ServerINI=\\192.168.1.1\USB DLM_share\usbdlm.ini

For this USB DLM.INI only section [NetworkLetters] is read!

The file is read from the context 'LocalSystem' of the computer which the USB DLM service is running on with null network credentials. So the share should have read access granted to 'Everyone' and furthermore the share must be allowed to be a null session share:

http://www.uwe-sieber.de/nullsessionshares_e.html

For network drives only an UNC path works here because on system startup the network drive letters are not yet created and furthermore they exist in the context of the interactive user only under XP and higher.

To avoid delays it's read on startup of the USB DLM service only.

Letters 1 to 26 can be used in addition.

Default Letters

In section [DriveLetters] you can configure default letters for USB drives. Since V5.0 there is no more section numbering required, so we talk here about the last section [DriveLetters].

Sample:

```
[DriveLetters]
Letter1=U
Letter2=V
Letter3=W
```

Or short

```
[DriveLetters]
Letters=U,V,W
```

Because there are no criterions in the section, only the default criterions are tested. These are:

```
BusType=USB
```

```
DriveType1=REMOVABLE
DriveType2=FIXED
```

This means that USBDLM by default deals with USB flash drives and USB hard drives only. It's for easier config, for compatibility with USBDLM V3.x and because of the program's name USB Drive Letter Manager...

You can make USBDLM read drive letters from an external INI file too, e.g. on the newly attached drive:

```
[DriveLetters]
Letters=%drive%\usbdlm.ini
```

%drive% is a USBDLM [variable](#) {linkID=} which it replaces with something like "X:".

Both ways can be used together, the external file has precedence:

```
[DriveLetters]
Letters=%drive%\usbdlm.ini
Letter1=U
Letter2=V
```

This makes USB DLM read drive letters from the USB DLM.INI on the drive first and then the others.

In the external INI file only one simple [DriveLetters] section is valid as shown in the first sample.

To make a section responsible only when the USB DLM.INI on the drive is present we need the **FileExists** [{linkID=69}](#) criterion:

```
[DriveLetters]
FileExists=%drive%\usbdlm.ini
Letters=%drive%\usbdlm.ini
```

Hint: Using an INI on the attached drive can cause delays because for reading the INI the drive's file system must be mounted. No problem with most drives but some don't like to be read too quickly after arrival... The maximum waiting time for the file being read is 10s and can be changed in section Settings like so:

```
[Settings]
VolumeInformationMaxWait=12000
```

Limiting the number of drives

If there is group policy defined for the drive letters configured in a DriveLetters section, e.g. which prevents programs started from these letters, then it should be ensured that no other letters are assigned in case the system is flooded with drives, e.g. by a 5 slot card reader.

This can be achieved by configuring "no letter" as the last one (a minus character):

```
; max 3 USB drives
[DriveLetters]
Letter1=U
Letter2=V
Letter3=W
Letter4=-
```

As list:

```
[DriveLetters]
Letters=U,V,W,-
```

No Media No Letter

The typical 20 in 1 card reader takes for each slot one drive letter, if you own a card of this type or not. Under XP one knows which drive letter is assigned to which slot. Since Vista at least the Windows Explorer hides empty slots even the drive letters stay assigned (or lets say wasted).

So, let USBDLM remove the drive letters and let it assign one when a card is inserted.

[Settings]

NoMediaNoLetter=1

To take effect the device in question must be reattached or the USBDLM service must be restarted.

NoMediaNoLetter=0 -> no drive

NoMediaNoLetter=1 -> multislot-cardreaders only

NoMediaNoLetter=2 -> singleslot-cardreader too

NoMediaNoLetter=3 -> all 'removable' drives

It does not work with floppy drives since there is no media insert notification.

There is no reliable procedure to identify "singleslot-cardreaders". USBDLM assumes a removable USB drive being a cardreader when its device name contains characteristic words as "reader", "card" and some others. If your card reader isn't detected as such then either set NoMediaNoLetter=3 or use section [NoMediaNoLetterInclude] (see below).

UsbDriveInfo shows if NoMediaNoLetter is active for each removable drive. If it is not then it shows what value is required for NoMediaNoLetter:

```
MountPoint      = T:\
Volume Label    = ---
Volume Size     = no media
Volume Serial   = ---
Volume Name     = \\?\Volume{e5d9c3d5-2df8-11e1-8cd3-001a4d539f5f}\
Partition Name  = \Device\Harddisk9\Partition1
Bus Type       = USB
Drive Type      = removable
NoMediaNoLetter = no (configure = 3)
```

Force devices being handled by the "no media no letter" function

Sample to force the NoMediaNoLetter feature for an old Iomega ZIP IDE drive and an ATAPI CDROM:

[NoMediaNoLetterInclude]

```
DeviceID1=IDE\DISKIOMEGA_ZIP_100
DeviceID2=IDE\CDROMHL-DT-ST_DVDRRW_GWA-4161B
```

On insertion of a media USB DLM assigns the first free letter. If you want a certain letter, it must be configured like this:

```
[DriveLetters]
DeviceID=IDE\DISKIOMEGA_ZIP_100
Letter=Z
```

```
[DriveLetters12]
DeviceID=IDE\CDROMHL-DT-ST_DVDRRW_GWA-4161B
Letter=L
```

Exclude devices from "no media no letter" function

To make the "NoMediaNoLetter" function work, the USB DLM service has to register for some notification messages. This is done for all drives types configured in the NoMediaNoLetter value. Some devices make trouble here, they appear to have no media when they are just attached or USB DLM get flooded by wrong removal messages.

If a device makes trouble then it can be excluded from being handled by NoMediaNoLetter by its device ID.

Sample to exclude the Apple iPod:

```
[NoMediaNoLetterExclude]
DeviceID1=ven_apple&prod_ipod
```

This is already build in because early iPods first appeared to have on media present and the iTunes software seems to need the iPod's drive letter.

Furthermore devices with "Iomega", "ZIP" or "JAZ" are excluded because I think it's just not appropriate for such floppy like drives but it can be overruled by using [NoMediaNoLetterInclude], see below.

"device ID" means an at least 8 characters long part of either the "Drive DevID" or "Ctrl DevID". The "Friendly Name" as shown by the UsbDriveInfo tool can be used too but the configured name must contain a space. Otherwise the "Friendly Name" might not be determined. Or use [Wildcards](#).

DeviceID1 to DeviceID9 and DeviceID can be configured.

Problem 1: Windows Portable Device driver does not start

Under Vista, Windows 7 and surprisingly Server 2008 for each USB drive Windows also installs a Windows Portable Device (WPD) driver. But the WPD driver relies on a drive letter for no reason. For drives without a drive letter their WPD driver fails with Code 10 (device cannot start). Ask Microsoft why.

This is not a problem with USB DLM, it is just because a drive having no drive letter assigned.

Problem 2: Drive no more shown in the "Safely remove hardware" list under Windows 7

The Safely remove hardware facility of Windows 7 does not like drives mounted into NTFS folders. This is again not a problem with USB DLM, it is just because a drive having no drive letter assigned.

Drive letters come back

If the drive letters appear insistently again, then another software is working which checks the drive letters and "repairs" them.

The "U3 launchpad" of U3 flash drives is known for doing so when started (the U3launch.exe on the fake CDROM drive).

This ignorant behaviour is present since U3 was introduced back in 2005.

Another software which is known to assign drive letters is "Secure Storage Device SDK" (SSDService.exe) from MXI and "TomTom MyDrive Connect".

To discover which process is changing the drive letters you can use the SysInternals ProcessMonitor:

<http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>

Set these filters:

```
Path    contains  DosDevices
Operation is      RegSetValue
```

In the toolbar leave the registry button pressed, the other three can be deselected.

When a drive letter gets assigned an RegSetValue access should be logged.

If it is the U3 Launchpad, then here are some Workarounds:

Workaround 1: Deactivating U3 by removing the drive letter of the faked CDROM drive

```
[DriveLetters]
DriveType=CDROM
DeviceType=U3
Letter=-
```

Workaround 2: No admin privileges for the U3 launchpad, then it cannot assign drive letters. So either work as restricted user or let the U3launch.exe start restricted by USB DLM, like this:

```
[OnArrival]
DriveType=CDROM
```

```
FileExists=%drive%\u3launch.exe  
open=%drive%\u3launch.exe  
restricted=1
```

Or if installed local, remove it from the Windows Startup and let USBDLM start it restricted:

```
[OnUserLogon]  
open="%ALLUSERSPROFILE%\Application Data\U3\U3Launcher\LaunchU3.exe"  
restricted=1
```

But the current version of the U3 launchpad V1.6 usually does not work without admin privileges.

Sometimes this procedure helps:

Remove all USB 'removable' drives. Attach the U3 drive while logged on as Admin. The U3 Launchpad can now batch off its strange installation orgy. Once successfully passed, the launchpad usually works now without admin privileges.

BTW: Don't put the U3 data drive on letter A: or B:, the launchpad isn't able to deal with these letters.

Workaround 3: let USBDLM "fight" against it

Actually I expected this U3 bug fixed sooner or later. But three years after the invention of U3 the launchpad it is still not able to distinguish between its own data drive and other USB removable drives. So, let's USBDLM fix it.

```
[Settings]  
FightU3Bug=1
```

When drive letter of USB removable drive appear unexpected and within the last 15 Seconds an U3 CDROM has been attached, the new drive letters are removed again. In the Windows Explorer there might be left one or more zombie drives with red question mark. An refresh by pressing F5 removes them.

If the U3 Launchpad is installed on the system drive, then it is started thru the Startup folder and does permanently assign drive letters to new USB "removable" drives. Here helps

```
[Settings]  
ForceNoMediaNoLetter=1
```

With this setting USBDLM removes drive letters again which are assigned to USB drives without a media present.

Static Letters

All letters defined in section [StaticLetters] are not touched by USBDLM. This way you can still assign certain USB drive letters by means of the Windows Disk Management which might be more handy than configuring something special in the USBDLM.INI.

Sample: Don't touch letters U to Z:

```
[StaticLetters]  
Letters=U-Z
```

Do not configure "low" letters (the first free ones) here because these are assigned by Windows to new drives and USBDLM wouldn't change them!

Drive Letters depending on Criteria

Since V4.0 you have complete freedom of using many different criteria.

You need then several sections [DriveLettersX], in which the different criteria are defined together with the target drive letters.

The sections are tested from [DriveLetters1] to [DriveLetters99] and finally under USBDLM V4 one [DriveLetters] and up to 100 [DriveLetters] since USBDLM V5. So you can still use numbered section as with USBDLM V4 or omit the section numbers.

The first section whose criteria fits to the drive is used. So, configure special cases first (or at low numbers) and more general ones later (or at higher numbers) and the most general last.

This works the same way for [autorun events](#) by means of sections [OnArrival], [OnRemoval], etc.

UsbDriveInfo is analyzing the USBDLM.INI from the folder of the USBDLM.EXE which is registered as a service. If this is not the current folder then it shows a warning. It shows for each drive copy&paste ready criteria lines and which section is used for them (press F5 after you changed the INI), so you can easily check if you have configured right.

This is (shortened) what UsbDriveShows for each drive:

```
===== Storage Volume on Disk 1 in Partition 1 =====
MountPoint      = Z:\
Volume Label    = King16Back
Volume Size     = 15.6 GB / 14.6 GiB / 15.619.129.344 Bytes
Volume Name     = \\?\Volume{09247b47-82df-11e4-b297-001d729ad2b2}\
DeviceID       = STORAGE\VOLUME_??_USBSTOR#DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP#...
DeviceID Drive = USBSTOR\DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP#001CC0C60DBDCC0C000010B&0
DeviceID Ctrl  = USB\VID_0951&PID_1693\001CC0C60DBDCC0C000010B
INI Sections   = [DriveLetters]:33, [OnArrival]:38

----- USBDLM Criteria -----
MinVolumeSize=14G
MaxVolumeSize=16G
MinDriveSize=14G
MaxDriveSize=16G
DeviceID=USBSTOR\DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP#001CC0C60DBDCC0C000010B&0
DeviceID=USB\VID_0951&PID_1693\001CC0C60DBDCC0C000010B
DeviceID=Kingston DataTraveler 111 USB Device
PortName=6-3
PortName=PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7-3
VolumeLabel=King16Back
VolumeSerial=A4605DC1
DriveType=REMOVABLE
BusType=USB
PartitionNumber=1
DiskSignature=C3072E18
```

"INI Sections" shows which sections are used for a volume, for instance [DriveLetters]:33 means section [DriveLetters] in line 33.

Default criteria are:

BusType=USB

and

DriveTypes=REMOVABLE,FIXED

This means that a section without criteria fits for all USB drives but not USB floppy drives and USB CDROM drives.

If a different BusType or DriveType is used, then the default value is not used anymore.

Untypical sample for showing how it works, criteria in **green**, target letters in **violet**:

For USB DLM V4 or V5:

```
[DriveLetters]
DeviceID1=USB\VID_067B&PID_2517
DeviceID2=USB\VID_1234&PID_5678
Letter1=X
Letter2=Y
```

For USB DLM V5 with unnumbered values:

```
[DriveLetters]
DeviceID=USB\VID_067B&PID_2517
DeviceID=USB\VID_1234&PID_5678
Letter=X
Letter=Y
```

This section is used if a drive with **either** DeviceID USB\VID_067B&PID_2517 **or** USB\VID_1234&PID_5678 is attached. The drive then gets assigned X: if available, otherwise Y:. If both are in use then the first free letter is assigned. If in this situation no letter shall be assigned an additional Letter=- can be configured.

There is no relation between the numbers!

Typical sample for getting these two drives on their own letters:

```
[DriveLetters]
DeviceID=USB\VID_067B&PID_2517
Letter=X
```

```
[DriveLetters]
DeviceID=USB\VID_1234&PID_5678
Letter=Y
```

If you want two drives handled by one section then use the same criteria two times:

```
[DriveLetters]
DeviceID=USB\VID_067B&PID_2517
DeviceID=USB\VID_1212&PID_8976
Letter=X
```

Here one of the device IDs must fit the drive.

More samples:

1. FireWire drives at F, USB drives above 10GB at U, all other USB drives at W or Z:

```
;FireWire at F:  
[DriveLetters]  
BusType=FireWire  
Letter=F
```

```
;USB >10GB at U:  
[DriveLetters]  
MinDriveSize=10GB  
Letter=U
```

```
;all other USB drives at W: or Z:  
[DriveLetters]  
Letter=W  
Letter=Z
```

2. USB multislot cardreader at L, M, N, O; all other USB drives at W or Z:

```
;MultiSlotCardReader at L:, M:, N: and O:  
[DriveLetters]  
DeviceType=MSCR  
Letters=L,M,N,O
```

```
;all other USB drives at W: or Z:  
[DriveLetters]  
Letters=W,Z
```

Multiple criteria of the same type can be used, then one of them must match.
If multiple types of criteria are used, one of each type must match.

Sample for USB and FireWire drives above 10GB at F:

```
[DriveLetters]  
BusType=USB  
BusType=FireWire  
MinDriveSize=10GB  
Letter=F
```

Sample for a CF-Card in a PCMCIA slot at P:

```
[DriveLetters]  
BusType=ATA  
DeviceID=PCI\VEN_1217&DEV_7135  
Letter=P
```

The real [device ID](#) is shown by UsbDriveInfo.

This way it does not work:

```
[DriveLetters]
DriveType=fixed
Letters=K,L
```

;special USB hard drive at I:

```
[DriveLetters]
DriveType=fixed
DeviceID=USBSTOR\DISK&VEN_WDC_WD20&PROD_00BB-00GUA0
Letters=I
```

Why? The first section 'catches' all USB harddrives, the section after is never reached.
So, always configure from special cases upwards to the general cases.

All the criteria are described in detail in the following sections.

Here is a list of all supported Criteria:

- MinVolumeSize Minimum size of the volume
- MaxVolumeSize Maximum size of the volume
- MinDriveSize Minimum size of the whole drive
- MaxDriveSize Maximum size of the whole drive
- DeviceID Device ID of the volume, the drive or its "Friendly Name"
- PortName Name of the USB port, e.g. 3-1
- VolumeLabel (*) Volume Label (also called Drive Name)
- VolumeSerial Serial number of the volume, e.g. 1234-A56E
- DriveType e.g. REMOVABLE, FIXED, CDROM, FLOPPY, RAMDISK
- DeviceType e.g. MSCR, TrueCrypt, BitLocker
- BusType e.g. USB, FireWire, ATA, ATAPI, SCSI
- PartitionNumber Partition number
- FileExists presence of a file or folder
- FileNotExists Absense of a file or folder
- DirExists presence of a folder
- DirNotExists Absense of a folder
- UserName User name of the logged on user
- UserGroup Group name of the logged on user
- UserIsAdmin 1 if user is an admin, 0 otherwise
- ComputerName Computer name, e.g. for a unique USBDLM.INI but different behaviour
- ProcessIsRunning A certain process is running
- ProcessIsNotRunning A certain process is not running

- MinOsVer / MaxOsVer Min or max version of the operating system

(*) May cause slower arrivals because the file system must be mounted. Therefore, if possible, put sections using such criteria to high section numbers. Sample:

Bad: Even for the drive with device ID `USB\VID_9876&PID_5432` the volume label is checked and therefore the file system mounted before USBDLM knows the desired drive letter:

```
[DriveLetters]
VolumeLabel=Backup1
Letter=U
```

```
[DriveLetters]
DeviceID=USB\VID_9876&PID_5432
Letter=X
```

Good: For the drive with device ID `USB\VID_9876&PID_5432` the volume label is not checked and therefore the file system is not mounted before USBDLM can change the drive letter.

```
[DriveLetters]
DeviceID=USB\VID_9876&PID_5432
Letter=X
```

```
[DriveLetters]
VolumeLabel=Backup1
Letter=U
```

Hint: Fictitious criteria do not work.

Another batch of criteria is related to the time, see [Letters by Time](#).

Drive Letters by Drive Size

The drive size can relate to the size of the whole disk or a single partition.
You can use units as KB, MB or GB, otherwise the unit is single bytes!
A KB means 1,000 Bytes, an MB 1,000,000 Bytes and so on.

A typical 40 GB drive usually has a size of a bit above 40 GB, a single partition on it usually a bit below 40 GB.
So it makes sense to use a bit lower values like 39 GB.

Only whole numbers are allowed.

A minimum or maximum size can be used, so available config items are:

MinVolumeSize
MinDriveSize
MaxVolumeSize
MaxDriveSize

Sample:

```
;USB drives above 40GB at U:  
[DriveLetters]  
MinDriveSize=40GB  
Letter1=U
```

```
;USB drives with including 10GB and more at X:  
[DriveLetters]  
MinVolumeSize=9GB  
Letter1=X
```

Because the author often used DiskSize instead of DriveSize (and wrote it into the help), DiskSize and DriveSize work both equally.

Letters by Device ID

When a device ID is configured in a section then there is no need to configure a BusType or DriveType because a device ID is definite enough.

The device ID is a bit hard to configure...

Sample for two special drives at U and another special one at X:

```
[DriveLetters]
DeviceID=USB\VID_067B&PID_2517
DeviceID=USB\VID_1234&PID_5678
Letter=U
```

```
[DriveLetters]
DeviceID=USB\VID_9876&PID_5432
Letter=X
```

The "Friendly Name" as shown by the UsbDriveInfo tool can be used too but the configured name must contain a space. Otherwise the "Friendly Name" might not be determined. Furthermore determining the friendly name sometimes fails, using a real device ID is more reliable.

```
[DriveLetters]
DeviceID=Corsair Flash Voyager
Letter=V
```

Sample for a SD reader in a laptop with BusType SCSI:

```
[DriveLetters]
DeviceID=PCI\VEN_1217&DEV_7120
BusType=SCSI
Letters=F,G
```

The BusType can be omitted here since V4.3 because a DeviceID is defined which is definite enough.

You can determine the device ID by means of the UsbDriveInfo tool.

As Device-ID you can use the DevID of the drive or of its controller or its "Friendly Name".
For USB drives the controller is the USB controller chip.
UsbDriveInfo shows the criteria ready for copy and paste to the INI:

----- USBDLM Criteria -----

```
MinVolumeSize=14G
MaxVolumeSize=16G
MinDriveSize=14G
MaxDriveSize=16G
DeviceID=USBSTOR\DISK&VEN_KINGSTON&PROD_DATA TRAVELER_111&REV_PMAP\001CC0C60DBDBCC0C000010B&0
DeviceID=USB\VID_0951&PID_1693\001CC0C60DBDBCC0C000010B
DeviceID=Kingston DataTraveler 111 USB Device
PortName=6-3
PortName=PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7-3
VolumeLabel=King16Back
VolumeSerial=A4605DC1
DriveType=REMOVABLE
DeviceType=DiskReadOK
BusType=USB
PartitionNumber=1
DiskSignature=C3072E18
```

An at least 8 characters long part from start is enough. Or use [Wildcards](#).

If you deal with a large number of drives then you can use an external file with the device IDs. This saves the numbering.

```
[DriveLetters]
DeviceIDs=C:\USBDLM_DeviceIDs.txt
Letter1=V
```

In the text file there is just one device ID per line. Comments are lead in by a semicolon, empty lines are allowed.

Sample:

```
;admin's drives
USB\VID_067B&PID_2517
USB\VID_1234&PID_56789

;chef's drives
USB\VID_9876&PID_5432
USB\VID_4567&PID_4321

;others
USB\VID_3456&PID_3543
USB\VID_3456&PID_1432
```

The string compare is done case insensitive. The whole configured string must be found in the drives device ID to give a match.

Using the USB hardware serial number

Windows puts the serial behind vendor and product ID in the string. In the sample above the USB controller's device ID was

```
"USB\VID_0951&PID_1693\001CC0C60DBDBCC0C000010B".
```

The USB hardware serial number is 001CC0C60DBDBCC0C000010B.

Meanwhile pretty rare, there are devices without a hardware serial number, here Windows generates one, in this case it contains at least two & chars. Windows sees such devices a new on every different USB port.

To make the device ID match the device on any USB port, just omit the last part like so:

```
DeviceID=USB\VID_0951&PID_1693
```

This also makes it fit for multiple samples of the USB drive model with unique serials.

For devices with serial you can use the whole device ID or just the serial because it appears in the device ID strings.

```
[DriveLetters]
DeviceID=001CC0C60DBDBCC0C000010B
Letter=V
```

If you configure then Letter1=-- for all other USB drives in a higher section or section [DriveLetters] then you have the one and only functional USB drive.
If USB floppy and CD drives shall be blocked too then:

```
[DriveLetters]
DriveType=floppy
DriveType=cdrom
DriveType=removable
DriveType=fixed
Letter=--
```

Or short:

```
[DriveLetters]
DriveTypes=floppy,cdrom,removable,fixed
Letter=--
```

Or even shorter:

```
[DriveLetters]
DriveType=any
Letter=--
```

Drive Letters by USB Port

Since V4.4 this can be used without hassle. No more delays. UsbDriveInfo shows now the USB port name determined by the old and by the new algorithm. Both should be identically. The only exception is that 1 port hubs (some active extension cables or embedded hubs and a very few flash drives) are no more ignored.

It's a bit hard to configure...

```
[DriveLetters]
PortName=3-2
Letters=U,R
```

;for multiple ports as the ports of an hub connected to 5-1

```
[DriveLetters]
PortName=5-1-1
PortName=5-1-2
PortName=5-1-3
PortName=5-1-4
Letters=X
```

Or short using [Wildcards](#):

;for multiple ports as the ports of an hub connected to 5-1

```
[DriveLetters]
PortName=5-1-?
Letters=X
```

First the physical USB ports have to be identified for the configuration. You get the port names by attaching a drive to each port to configure. Then run the UsbDriveInfo tool. UsbDriveInfo will show something like this for each USB drive:

```
MountPoint      = Z:\
Volume Label    = King16Back
Volume Size     = 15.6 GB / 14.5 GiB / 15.619.129.344 Bytes
Volume Serial   = A460-5DC1
FileSystem BS   = NTFS
Volume Name     = \\?\Volume{09247b47-82df-11e4-b297-001d729ad2b2}\
KernelName     = \Device\HarddiskVolume21
Partition Device = \\.\Harddisk1Partition1
DeviceID       = STORAGE\VOLUME_??_USBSTOR#DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP#...
DeviceID Drive = USBSTOR\DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP#001CC0C60DBBCC0C000010B&0
DeviceID USB   = USB\VID_0951&PID_1693\001CC0C60DBBCC0C000010B
Device ID HostCtl = PCIVEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7
USB Friendly Name = Kingston DataTraveler 111
USB Port Name   = 3-2
INI Sections    = [DriveLetters]:33
```

For this drive 3-2 is the setting for "PortName" we need. If there is a USB hub in between then a typical port name is 3-2-1, with two hubs 3-2-1-1 and so on. USB hubs with 7 ports can be in fact two cascaded 4 port hubs, so 4 of the 7 ports get an additional step in the USB tree.

USB3 controllers usually have separate logical ports for USB 1+2 and 3 devices, so the port number differs when a USB2 or a USB3 devices is attached to the same physical port, see here: http://www.uwe-sieber.de/usbtreview_e.html#USB3

But this is true for ports of root hubs only. Ports of normal USB hubs have one incarnation only.

Sample to have a fixed letter for a USB3 port, no matter if a USB2 or 3 device is attached:

```
[DriveLetters]
PortName=3-2
PortName=3-5
Letters=X
```

If there are multiple Windows installations on a computer then the USB Port names will be equal only for identical Windows Versions.

Of course, if you change your USB hub assignments, you'll have to do all this over, too.

Furthermore the numbers of the Root Hubs are not constant when Host Controllers are removed or added. The numbering Windows does is not predictable.

If you have a removable USB Controller and the internal root hubs change their number when it is removed or added then an alternative notation can be used. Instead of the Root Hub's number use its host controller's device ID which is a constant.

In the sample above it is

```
PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7
```

The resulting "long" port name would be here

```
PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7-2.
```

If each type of host controller is present only once then the ID can be shortened using wildcards. Then it would work across multiple Windows installations on the same computer:

```
[DriveLetters]
PortName=PCI\VEN_8086&DEV_283A&*-2
Letters=X
```

Non-USB drives

Since V4.7 this is no more limited to USB drives. If a drive has a "location" string like "Channel 1, Target 0, Lun 0" then USB DLM generates a short version of it by picking the numbers and putting them in a short string like 1-0-0.

Alternatively you can use the "location" string as shown by UsbDriveInfo, also using wildcards.

Such a location string is usually found for ATAPI, SCSI and SATA drives.

Remember that you have to configure a [BusType](#) line too because we deal with non USB drives here!

Sample for an ATAPI CD/DVD drive on letter X:

```
[DriveLetters]
```

DriveType=CDROM
BusType=ATAPI
PortName=Bus Number 0, Target ID 1, LUN 0
Letters=X

Or with the short notation for the port:

[DriveLetters]
DriveType=CDROM
BusType=ATAPI
PortName=0-1-0
Letters=X

Drive Letters by USB Speed

Since V5.0 the USB connection speed and the maximum speed of a USB drive can be use as criteria like so:

```
[DriveLetters]
UsbSpeed=High-Speed
Letters=H
```

```
[DriveLetters]
UsbSpeed=SuperSpeed
Letters=S
```

Probably it is most useful for [customized balloon tips](#):

; USB SuperSpeed drive which is connected at High-Speed only:

```
[BalloonTipsOnArrival]
UsbSpeed=Low-Speed
UsbSpeed=Full-Speed
UsbSpeed=High-Speed
UsbMaxSpeed=SuperSpeed
Title=%FriendlyName% %BusType% %UsbMaxSpeed% Device at %UsbSpeed% - %DriveSize%
Text=mounted to %Root% ( %FsName% )
```


Drive Letters by Volume Label

The Volume Name (also called Drive Name) is the text shown in the drive's properties dialog (right click the drive, "Properties"). The label shown in the Windows Explorer can be the volume label but it also can be a setting from the Windows Registry. If you change the label by single clicking the drive or by pressing F2, it depends on where the current label comes from, which one is changed.

To see and change the real volume label, right-click the drive in the Windows Explorer and select Properties.

To be really sure that the real volume label is changed, use the LABEL command on the Windows command prompt.

Sample:

```
[DriveLetters]
VolumeLabel=Monday
VolumeLabel=Wednesday
VolumeLabel=Friday
Letter=U
```

```
[DriveLetters]
VolumeLabel=Tuesday
VolumeLabel=Thursday
VolumeLabel=Saturday
Letter=X
```

Drives named Monday, Wednesday or Friday get U and drives named Tuesday, Thursday or Saturday get X.

You can use [Wildcards](#):

```
[DriveLetters]
VolumeLabel=Backup?
Letter=U
```

This would fit for a VolumeLabels Backup1, Backup2 etc.

With a minus char you can test of an empty volume label:

```
[DriveLetters]
VolumeLabel=-
Letter=U
```

Using as drive letter

Since V4.6 USB DLM can use the volume label directly for the drive letter. For this purpose there is the new USB DLM variable %LetterFromLabel%.

For LetterFromLabel a letter is extracted from the drive's volume label by looking for:

- the last letter which is followed by a colon and preceded by a non-letter, e.g. "Drive X:" or "X: Drive"
- the last letter which is surrounded by non-letters, e.g. "Drive X", "Drive_X", "Drive_X_", "Drive (X)", "X Drive"

Non-letters are the begin, the end, the space and the characters !"#\$%&'()*+,-./:;<=>?@[\\]^_`{|}~«».

If nothing fitting is found then the variable is empty.

Remark: The colon is a valid character in volume labels of NTFS formatted drives only.

Sample for always trying to extract a letter from the label and fall back to Z: if there is none:

```
[DriveLetters]
Letter1=%LetterFromLabel%
Letter2=Z
```

To avoid unintentional interpretation as a drive letter, a VolumeLabel criteria with [Wildcards](#) can be configured, to narrow down the usage of a section to exactly the desired naming scheme.

Sample: Use the section only if the label ends on underline plus one character (e.g. Drive_X):

```
[DriveLetters]
VolumeLabel=*_?
Letter1=%LetterFromLabel%
Letter2=Z
```

Here some wildcards samples for the VolumeLabel criteria:

- *_? begins with any (or none) character, ends on underline, followed by exactly one character; Sample: "Drive_X"
- * ? begins with any (or none) character, ends on space, followed by exactly one character; Sample: "Drive X"
- *?: begins with any (or none) character, ends on space plus one character plus colon; Sample: "Drive X:"
- ?:* begins with exactly one character plus colon plus space, followed by any (or none) character; Sample: "X: Drive"
- *(?) begins with any (or none) character, ends on one character enclosed by parenthesis; Sample: "Drive (X)"

Of course you can use more specific labels, like "USB_W", "USB_X", "USB_Y" and the criteria "VolumeLabel=USB_?", to avoid unexpected results with foreign drives:

[DriveLetters]

VolumeLabel=USB_?

Letter1=%LetterFromLabel%

Letter2=Z

Using the volume label may cause slower arrivals or will not work at all because the file system must be mounted to read it. Some drives need a while for getting ready.

The maximum waiting time is 10 Seconds and can be configured in Milliseconds globally in section Settings:

[Settings]

VolumeInformationMaxWait=10000

Drive Letters by Volume Serial Number

The Volume Serial Number a 32 bit random value, created when a drive is formatted. Since it cannot be changed by the user offhand it might be a better choice as criteria than the Volume Label. Furthermore determining it does not cause delays because USB DLM reads it directly from the Disk device instead from the volume's file system.

Determine the serial:

Start -> Run
CMD

```
C:\>dir u:\*.x
Volume in drive U is TEST
Volume Serial Number is B92B-8000
```

'*.x' is just a file name pattern which usually results in zero hits.

The UsbDriveInfo tool shows the serial too:

```
===== Storage Volume on Disk 1 in Partition 1 =====
MountPoint      = U:\
Volume Label    = TEST
Volume Size     = 15.6 GB / 14.5 GiB / 15.619.129.344 Bytes
File System     = NTFS 3.1, 4 KB clusters, 145 MB free
Volume Serial   = B92B-8000
Filesystem BS   = NTFS
```

Sample:

```
[DriveLetters]
VolumeSerial=B92B-8000
Letter=U
```

```
[DriveLetters]
VolumeSerial=C8D3-F123
Letter=V
```

Or several volumes with different serials on one letter (one after the other):

```
[DriveLetters]
VolumeSerial=B92B-8000
VolumeSerial=34AB-E831
VolumeSerial=A34C-3080
Letter=U
```

The volume serial can be changed by means of the command-line tool [VolumeID from SysInternals](#).

This way you can create drives with consecutive serials and then use wildcards:

[DriveLetters]

VolumeSerial=BABA-00??

Letter=B

Drive Letters by Drive Type

The following drive types are defined in USBDLM:

FLOPPY Floppy drives
REMOVABLE non floppy drives with removable media as USB flash drives and card readers
FIXED Hard drives
CDROM CD / DVD / Blu-ray drives
UNKNOWN unknown, some card readers in notebooks
RAMDISK RAM drives (many RAM drives appear as FIXED, check with UsbDriveInfo)
ANY / ALL all types (**use it carefully!**)

Sample for USB CDROMs at R:

```
[DriveLetters]
DriveType=CDROM
Letters=R
```

Sample for ATAPI-CD-ROMs at L: and M:

```
[DriveLetters]
BusType=ATAPI
DriveType=CDROM
Letters=L,M
```

If no DriveType is configured then the default types are used. These are REMOVABLE or FIXED.

If all drive types shall be valid for a section, then you can use "ALL".
Sample to get USB drive letters for members of the admins group only:

```
[DriveLetters]
UserGroup=Administrators
Letters=X,Y,Z
```

;no USB drives for all others

```
[DriveLetters]
DriveType=ALL
Letter=--
```

Do not configure like this:

```
;all others
[DriveLetters]
```

BusType=**ALL**
DriveType=**ALL**
Letter1=-

This would remove the drive letters of **all** drives, USB or not, because of BusType=ALL.

See [Letters by active User](#) for more information.

Letters by Device Type

The device type is something USBDLM specific to give some special devices a special handling.

Defined types are:

- MSCR -> Multislot-Cardreader (is always a CardReader too)
- CardReader -> drives with characteristic strings like 'Card' or 'Reader' in the device name
- U3 -> U3 flash drive or a harddisk with virtual CD-ROM drive e.g. for installing encryption software
- Floppy -> Floppy drive
- VirtualDrive -> all virtual drives as Subst, Network, TrueCrypt and other drives which are not full blown volumes
- SubstDrive -> Subst drive
- DokanDrive -> Virtual drive created by Hiroki Asakawa's user-mode file system library
- TrueCrypt -> a drive that is used as [TrueCrypt](#) container (or as container for any other encryption software, e.g. [FreeOTFE](#)), in fact volumes with random data in the first sector
- ReadSharingViolation -> read access to the volume caused ERROR_SHARING_VIOLATION, usually an already mounted TrueCrypt container
- TrueCryptVolume -> a mounted TrueCrypt volume (only supported for [OnArrival]/[OnRemoval] events, not for drive letters!)
- VeraCryptVolume -> a mounted VeraCrypt volume (only supported for [OnArrival]/[OnRemoval] events, not for drive letters!)
- OtfVolume -> a mounted OTFE volume (only supported for [OnArrival]/[OnRemoval] events, not for drive letters!)
- BitLocker -> a [BitLocker](#) volume
- AllZero -> first sector contains zeros only (unformatted or non Windows drives)
- ReadError -> Read data from the drive failed
- UnrecognizedVolume -> Windows says ERROR_UNRECOGNIZED_VOLUME, e.g. Ext3 volume or exFAT on XP without [KB955704](#) installed
- WindowsRecovery -> Windows Recovery partition (partition type 0x27 or PARTITION_MSFT_RECOVERY_GUID on GPT)

VirtualDrive, SubstDrive, DokanDrive, TrueCryptVolume, VeraCryptVolume and OtfVolume are relevant for virtual drives only, so they make no sense in DriveLetter sections.

The UsbDriveInfo tool shows the types.

Sample for multislot cardreader at L, M, N, O

```
[DriveLetters]
DeviceType=MSCR
Letters=L,M,N,O
```

Sample for multislot cardreader in an NTFS folder using the device name:

```
[DriveLetters]
DeviceType=MSCR
Letter1=C:\CardReader\%DevName%
```

It creates there one subfolder for each slot using the device name of the slot. Better card readers have 'talking' drive names like 'USB2.0 CF_MD', 'USB2.0 SD_MMC' and so on. Slots of no name readers have all the same, non talking name like 'USB USB', 'Generic USB', 'General Flash Disk'... USBDLM removes the folders when the card reader is disconnected.

A multislot cardreader is seen as such if it has three or more independent slots. Furthermore devices with two slots and 'reader' or 'card' and 'flash' in the device name.

Non-USB reader often seen in notebooks are not seen as MSCR. Configure it using the device ID then.

Sample for a SD reader in a laptop with BusType SCSI:

```
[DriveLetters]
DeviceID=PCI\VEN_1217&DEV_7120
BusType=SCSI
Letters=F,G
```

The BusType can be omitted here since V4.3 because a DeviceID is defined which is definite enough.

Sample to remove the drive letter of drives that are used as [TrueCrypt](#) container:

```
[DriveLetters]
DeviceType=TrueCrypt
Letter1=-
```

Sample to hide U3 AutoRun CD-ROMs:

```
[DriveLetters]
DriveType=CDROM
DeviceType=U3
Letter1=-
```

U3 is a standard for USB drives which allows to have all programs, data and settings on this drive and work with it on any computer without leaving any traces there.

For that there is the "U3 launchpad" on the drive which shall start automatically when the drive is attached. But Windows executes the open= line in the autorun.inf on CD drives only. Therefore these U3 drives contain a fake CD-ROM drive with the autorun part. This is the "U3 autorun drive" mentioned above.

If you don't need this, then you can configure an NTFS folder for it. Then you get rid of its autorun and you save a drive letter. Or just configure nothing in this section and remove the drive letter in the Windows disk management.

To remove the U3 functionality completely and irreversibly then use the U3 uninstall tool:

<http://u3.com/uninstall>

USB DLM sees all USB devices as U3 which have a CDROM and a data drive. This can be the case too on drives with encryption software or on promotion thumb drives.

Meanwhile you find such nasty CD-ROM drives on WLAN sticks and printers too for selfinstalling drivers... USB DLM does not recognize such drives as fake CD-ROMs so far.

Drive Letters by Bus Type

BusType means the way a drive is attached to the computer, like USB, FireWire, SCSI etc.

If no BusType is configured then the default BusType=USB is used. This means that by default a section fits for USB drives only. For non USB drives a BusType must be configured. The only exception is when a [DeviceID](#) is configured, then the BusType can be omitted even for non USB drives.

UsbDriveInfo shows the BusType of all drives but it shows non-USB drives only if in menu Options "Show all drives" is checked.

The BusTypes are:

USB
FireWire
SCSI
ATAPI
ATA
Remote
SSA
Fibre
RAID
iSCSI
SATA
SAS
SD (SD cardreader)
MMC (MMC cardreader)
Virtual (since Windows 7)
FileBackedVirtual (since Windows 7, mounted VHD files)
Unknown (some internal notebook cardreaders, virtual drives, legacy floppies)
ANY or ALL (any BusType - **use it carefully, it might remount your internal drives!**)

SATA drives often have the BusType ATA, SCSI or sometimes RAID... The driver of the drive's controller is responsible for reporting the bustype.

UsbDriveInfo shows the BusType:

```
MountPoint      = Z:\
Volume Label    = Backup_18
Volume Size     = 200 GB (NTFS)
Volume Serial   = 44E0-DE33
Disk Size       = 200 GB
Volume Name     = \\?\Volume{49e8598c-b0d0-11de-b41f-00304884ff2a}\
Partition Name  = \Device\Harddisk2\Partition1
Bus Type        = RAID
Drive Type      = fixed
INI Sections    = [DriveLetters], [OnArrival]
```

Use bus types with care, if you configure nonsense then you will get it!

For internal drives it's better to configure an additional DeviceID because it's unique. Before letting USBDM work with the INI let UsbDriveInfo show which sections are used for the drives, see the items of "INI Sections".

Sample for USB and FireWire drives larger than 100GB at X:

```
[DriveLetters]
BusType=USB
BusType=FireWire
MinDriveSize=100GB
Letter1=X
```

;alternative notation with comma

```
[DriveLetters]
BusTypes=USB,FireWire
MinDriveSize=100GB
Letter1=X
```

Sample for ATA disk with DeviceID 'IC35L120AVV207-0' at X:

```
[DriveLetters]
BusType=ATA
DeviceID=IC35L120AVV207-0
Letter=X
```

Sample for internal laptop cardreader with BusType SCSI:

```
[DriveLetters]
BusType=SCSI
DeviceID=SCSINDISK&VEN_O2MICRO&PROD_SD
Letter1=X
```

Since V4.3 the BusType can be omitted in the two upper samples since the DeviceID is usually definite enough.

Some internal laptop cardreaders have a variant BusType depending on the inserted card (SD or MMC). Because there is usually exactly one of these in the system, the device ID can be omitted:

```
[DriveLetters]
BusType=SD
BusType=MMC
Letter=X
```

Sample for virtual CD drives created by [Elby CloneDrive](#) or by the [Daemon-Tools](#) V3 and V4 on V, W, X, Y:

[DriveLetters]
BusType=SCSI
DeviceID=ELBY CLONEDRIVE
DeviceID=SCSI\CDROM&VEN_GENERIC&PROD_DVD-ROM
DeviceID=SCSI\CDROM&VEN_DAEMON
Letters=V-Y

In OnArrival/OnRemoval sections for volumes with the [DeviceType](#) TrueCryptVolume the BusType is not checked if no BusType is configured in a section, so the otherwise required BusType=unknown can be omitted then.

Drive Letters by Partition Number

Logical drives on hard disks have a partition number, the UsbDriveInfo tool shows them as part of the Partition Name:

```
MountPoint      = E:\
Volume Label    = Backup
Volume Size     = 18 GB (NTFS, 4 KB clusters, 3.5 GB free)
Volume Serial   = 78B8-C37E
Disk Size       = 100 GB
Volume Name     = \\?\Volume{770f7510-bce2-11df-8471-806d6172696f}\
Partition Name  = \Device\Harddisk0\Partition2
```

And ready for copy&paste among the criteria:

```
----- USBDLM Criteria -----
MinVolumeSize=17G
MaxVolumeSize=19G
MinDriveSize=99G
MaxDriveSize=101G
DeviceID=USBSTOR\DISK&VEN_GEN&PROD_GEN&REV_001\1234567890AB&0
DeviceID=USB\VID_04B4&PID_6830\1234567890AB
DeviceID=Kingston DataTraveler 111 USB Device
PortName=6-3
PortName=PCIVEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7-3
VolumeLabel=Backup
VolumeSerial=78B8C37E
DriveType=FIXED
BusType=USB
PartitionNumber=2
```

Sample for hiding partition number 2 on a USB hard drive with DeviceID
USB\VID_04B4&PID_6830\1234567890AB:

```
[DriveLetters]
DeviceID=USB\VID_04B4&PID_6830\1234567890AB
PartitionNumber=2
Letter1=-
```

Do not configure Letter=- because this would remove the whole device!

The PartitionNumber is checked on Fixed and Removable drives only. On Removable drives the number is always 1 or 0 because Windows since Windows 2000 does not support multiple partitions on 'removable' drives.

Letters by Disk Signature

Every drive which has a partition table has a 32 bit long signature. If zero then Windows asks if the disk shall be "initialized" when opening the Disk Management.

It's a random value which is used for identifying drives, e.g. it is part of the UniqueID used by the Windows Volume Manager.

Since V4.7.1 USB DLM can use the signature as criteria. It might be helpful if drives have no other differences.

The value is interpreted as little endian ULONG value, so if you find on the disk offset 1B8h the bytes F5 17 24 4D then the hex value is 4D2417F5.

UsbDriveInfo shows the signature like this along with the other disk properties:

```
Disk Signature = 4D2417F5 (F5,17,24,4D)
```

Sample for having this drive on Z:

```
[DriveLetters]
DiskSignature=4D2417F5
Letter=Z
```

The signature must be noted with eight digits or with wildcards.

On GPT drives the disk signature is usually null, here the Disk ID Guid can be used instead, like so:

```
[DriveLetters]
GptDiskIdGuid={ac6ce627-1b3a-4b9d-9383-94842785585d}
Letter=Z
```

UsbTreeView shows it in the disk's properties.

Letter by Existence of a File or a Folder

```
[DriveLetters]
FileExists=%Drive%\drive_a.id
Letter1=A
```

This sample works for USB drives only because of the default value BusType=USB.

To make it work with other types too, just configure them as additional BusTypes:

```
[DriveLetters]
BusType=USB
BusType=FireWire
BusType=SCSI
FileExists=%Drive%\drive_a.id
Letter1=A
```

Since V4.2 wildcards can be used. The evaluation is done by Windows, therefore it's different from the [USB DLM internal wildcards](#).

Only the given folder is scanned, no sub-folders!

Sample to start the "ImageViewer" if a JPG file or the folder "DCIM" (as created by all digicams) is found and the "VideoPlayer" for AVI and MPG:

```
[OnArrival]
FileExists=%Drive%\*.jpg
FileExists=%Drive%\DCIM
open="%ProgramFiles%\ImageViewer\ImageViewer.exe" %Root%
```

```
[OnArrival]
FileExists=%Drive%\*.avi
FileExists=%Drive%\*.mpg
FileExists=%Drive%\*.mpeg
FileExists=%Drive%\*.mkv
FileExists=%Drive%\*.mov
open="%ProgramFiles%\VideoPlayer\VideoPlayer.exe" %Root%
```

```
[OnArrival]
DirExists=%Drive%\VIDEO_TS
open="%ProgramFiles%\DvdPlayer\DvdPlayer.exe" %Root%
```

Since V4.3 USB DLM can test for absence of a file or folder (File**Not**Exists).

Sample to mount a TrueCrypt volume on X: if X: is available:

```
[OnArrival]
DeviceType=TrueCrypt
FileNotExists=X:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /v %PartitionName% /IX /q
```

Since V4.3 the presence of a folder can be checked explicit using DirExists and DirNotExists.

```
[OnArrival]
DeviceType=TrueCrypt
DirNotExists=X:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /v %PartitionName% /IX /q
```

Using any of these may cause slower arrivals because the file system must be mounted to check for a file or a folder.

Since V4.6.9.2 the maximum waiting time can be configured in Milliseconds. Default is 6.5 seconds for CD/DVD drives and two seconds for all other types. Sample for five seconds:

```
[DriveLetters]
FileTimeout=5000
FileExists=%Drive%\drive_a.id
Letter1=A
```

FileTimeout is used for all file criteria in the section.

Drive Letters by active User

The active user can be configured using its user name or the name of a group he is a member of.

To use groups defined on a server the USBDLM service has to start itself in the context of the logged on user. This USBDLM instance then determines the groups and reports them to the USBDLM service.

Use server groups only if required. To activate reading server groups set:

```
[Settings]
NetUserGroups=1
```

Since V4.7.2.50 beta this is probably no more required.

Sample for the user name:

```
;drive letters for the parents
[DriveLetters]
UserName=Homer
UserName=Marge
Letters=X,y
```

```
;no drive letters for the kids
[DriveLetters]
UserName=Bart
UserName=Lisa
UserName=Maggie
DriveType=ANY
Letter1=-
```

DriveType=ANY is required to remove the drive letters of USB floppy and USB CD/DVD drives too.

Sample for group names:

```
[DriveLetters]
UserGroup=Administrators
Letters=X,Y
```

```
;no USB drives for all others
[DriveLetters]
Letter=-
```

Or reverse:

```
;no drives for normal users  
[DriveLetters]  
UserGroup=Users  
UserGroup=Guests  
Letter1=-
```

```
;all other users can  
[DriveLetters]  
Letters=X,Y
```

Or using the user's admin state

```
[DriveLetters]  
UserIsAdmin=1  
Letters=X,Y
```

```
;no drives for non admins  
[DriveLetters]  
UserIsAdmin=0  
DriveType=ALL  
Letter=-
```

Under Vista/Win7+ for an Admin running under UAC UserIsAdmin is assumed as 1.

To make a certain USB drive work even a non admin is logged on, put a section with the drive's device ID in between or before both, see [Letters by Device ID](#).

```
[DriveLetters]  
DeviceID=USB\VID_067B&PID_2517  
Letters=X,Y
```

You can use [Wildcards](#). For instance admin* is good for Administrator and Admin.

While no user is active, UserName and UserGroup are empty, therefore no section which has UserGroup or UserName defined will be ever used then. UserIsAdmin is assumed as 0 while no user is active.

To make a section valid for "no user logged on" UserName=- or UserGroup=- can be configured.

While Windows shows the logon screen after switching the user, the last user is still logged on.

Letters by Computer Name

If multiple computers shall get the same configuration but act different, then the Computer Name can be used as criteria.

```
; USB drive on the teacher's computer on X, Y  
[DriveLetters]  
ComputerName=TeachersComputer  
ComputerName=AssistantsComputer  
Letters=X,Y
```

```
;no USB drives for all other computers  
[DriveLetters]  
DriveType=ANY  
Letter=-
```

DriveType=ANY is required to block USB floppy and CD drives too.

Letters by running Process

Probably useful in OnArrival or OnRemoval sections only...

```
; USB drives on U: if the Windows Calculator is running  
[DriveLetters]  
ProcessIsRunning=CALC.EXE  
Letters=U
```

```
; USB drives on V: if the Windows Editor is not running  
[DriveLetters11]  
ProcessIsNotRunning=NOTEPAD.EXE  
Letters=V
```

```
; others on Z:  
[DriveLetters]  
Letter=Z
```

Process names must be noted without path but with extension, as shown in the Windows TaskManager's "Processes" tab.

Letters by Operating System Version

If you want to have a unique configuration on many computers but a different handling depending on the operating system, then the criteria MinOsVer and MaxOsVer can be used.

Sample: Remove the drive letter of BitLocker To Go encrypted drives and show a warning on systems before Windows 7:

```
[DriveLetters]
MaxOsVer=6.0
DeviceType=BitLocker
Letter=-
```

```
[OnArrival]
MaxOsVer=6.0
DeviceType=BitLocker
open="%usbdlmpath%\usbdlm.exe" -balloon -time=20000 -title="%FriendlyName%" -text1="Cannot read
BitLocker encrypted volume" -icon=
```

The internal Windows version numbers are:

- Windows 2000: 5.0
- Windows XP: 5.1
- Windows XP x64: 5.2
- Windows Server 2003: 5.2
- Windows Vista: 6.0
- Windows Server 2008: 6.0
- Windows 7: 6.1
- Windows Server 2008 R2: 6.1
- Windows 8: 6.2
- Windows Server 2012: 6.2
- Windows 8.1: 6.3
- Windows Server 2012 R2: 6.3
- Windows 10: 10.0

Or just start UsbDriveInfo, it shows the version too.

Letters by Time

1. Daytime

; USB drives from 8 to 12 and 13 to 18 o'clock at U; no letter otherwise

[DriveLetters]

MinDaytime1=08:00

MaxDaytime1=12:00

MinDaytime2=13:00

MaxDaytime2=18:00

Letters=U

[DriveLetters11]

Letters=-

The time compare is implemented by text compare, so you have to note with two digits, e.g. 08:00, not 8:00. Furthermore a 24 hours notation is used, no am and pm!

2. Weekday

; USB drives from Monday to Friday at U; no letter otherwise

[DriveLetters]

Weekdays=MO,TU,WE,TH,FR

Letters=U

[DriveLetters11]

Letters=-

The weekdays are noted with English two letter abbreviations: MO, TU, WE, TH, FR, SA, SU. Ranges as MO-FR are not implemented.

3. Time since a system event

Supported events:

Boot

UserLogon

UserLogoff

UserLock

UserUnlock

UserDisconnect

UserConnect
UserActivated
UserDeactivated
SleepRequest
Sleep
Resume
Dock
Undock
UndockRequest
UndockCanceled

Each one noted after a MinTimeSince or MaxTimeSince, e.g. MaxTimeSinceBoot and the time in milliseconds.

If an event hasn't been triggered so far then both, Min and MaxTimeSinceXxx give no hit, but on startup of the USBDLM service the times are initialized in a meaningful way, e.g. UserDock is initialized with 1 if the computers is docked on startup.

Sample: On start of the USBDLM service convert ramdisk drive R: into the NTFS file system, but only if less than one minute since the system started (otherwise USBDLM has probably been started manually, so converting again makes no sense):

```
[OnServiceStart]
MaxTimeSinceBoot=60000
open="%windir%\System32\convert.exe" R: /fs:ntfs
system=1
```

Sample: Start the Windows Calculator on Resume, but only if the Sleep event it at least one minute in the past (otherwise the computer probably woke up again immediately unintended):

```
[OnResume]
MinTimeSinceSleep=60000
open=calc
```


Other criteria

Docking Status

IsDocked is 1 when a mobile computer is docked in its docking station.

Sample:

```
; USB drives on U if docked, on X otherwise
```

```
[DriveLetters]
```

```
IsDocked=1
```

```
Letters=U
```

```
[DriveLetters11]
```

```
IsDocked=0
```

```
Letters=X
```

Sleeping Status

Of course there are no events while a computer is sleeping. This is made for one special situation: When XP is sent to sleep the current user is not locked. The locking happens immediately after it wakes up again. To distinguish between a manual lock and the lock that happens after wakeup the criteria **IsSleeping** is made. This works because the lock notification comes before the resume notification. So, there is a short moment when IsSleeping is true.

Sample:

```
; turn off the screen when the computer is manually locked
```

```
[OnUserLock]
```

```
IsSleeping=0
```

```
open="C:\Tools\nircmd.exe" monitor off
```

NirCmd is Nir Sofer's great command-line tool for performing useful tasks without displaying any user interface.

<http://www.nirsoft.net/utils/nircmd.html>

Priorities

Up to V3.x the priorities of the sections were fixed. With version 4 you have a free hand.

USBDM always checks from [DriveLetters1] to [DriveLetters99] and finally [DriveLetters].
The first section that fits to the drive is used (only this one). So configure from the special cases upwards to the more general.

If the drive got the letter of a network share, then USBDM remounts it to the next available letter.
No need to configure anything for this function. If no letter is available then the drive is unmounted.

Finally, if the drive has no mount point at all, USBDM mounts it to the next available letter if there is one.

AutoRun

Windows AutoRun

The Windows AutoRun facility is not everybody's taste: Silly questions, mindless searching, tendency to it's own live and just not working...

So, just deactivate it, best using my tool [AutoRunSettings](#).

Microsoft's TweakUI for XP can do it too on first view but it cannot deactivate AutoRun for hard drives and it activates AutoRun for network and unknown drives because it completely ignores the Windows default values.

USB DLM's AutoRun functions

USB DLM has two mechanisms to start programs on arrival of a drive and on insertion of a media (card, disk, CD/DVD), see below.

By default the command is executed in the context of the active user. If there is no user active, the command-line is not executed, except there is system=1 configured for this open line.

Remote user are considered since V4.4.3, but only one. Multiple active users are available on servers only.

If there are more than one active user then USB DLM selects one depending on the setting "RemoteSessions" in section [Settings]:

RemoteSessions=0	remote sessions are ignored
RemoteSessions=1	remote sessions are considered when there is no active local user (default)
RemoteSessions=2	local and remote users are handled as equal

If there are multiple equal users then the user "wins" which logged on as last one.

In contrast to Windows' AutoRun, USB DLM's AutoRun works too if the drive is mounted into an NTFS folder or if it got no mount point at all.

By holding down the Shift key the execution can be skipped. Since Vista this works only if Force=0 is configured in the section because determining the keyboard status take some effort here.

Since V4.8.8 USB DLM supports opening document files with their associated application.

E.g.

```
open=%drive%\test.txt
```

would open the text file in the text processor.

If the path contains spaces and command-line parameters are used, then the path must be quoted. Sample:

```
open="C:\Program Files\AntiVir\AntiVir.exe" %drive% /scan
```

This is true too when environment variables are used which lead to spaces when expanded. Sample:

```
open="%ProgramFiles%\AntiVir\AntiVir.exe" %drive% /scan
```

With SafeCommandLines=1 in section [Settings] this is required whenever the command-line contains a space character. Sample

```
open=C:\Tools\test.exe -t
```

Seems obvious but could also mean an "C:\Tools\test.exe -t.exe". Therefore better always quote the path of the executable:

```
open="C:\Tools\test.exe" -t
```

By default the root of the drive is used as working directory. By means of a line workdir=xxx a different one can be specified.

Since V4.6 executable files are searched first in the Windows search path (environment variable PATH), then in the working directory.

This has been changed for security reasons: If for instance there is configured open=explorer %drive% then all a bad guy has to do is to put a bad explorer.exe on a USB drive and attach it...

An additional search path can be configured, e.g.

```
[Settings]
```

```
OpenSearchPath=C:\BatchFiles
```

AutoRun events on insertion or removal of a cardreader's media configured for the first time may work after reattaching the device or after a restart of the USB DLM service only. This is because for receiving these notifications USB DLM must register for them and this is done only if required. The decision if it is required or not is done only on arrival of a drive and on USB DLM's startup.

The [USB DLM variables](#) can be made available to the executed program as environment variables. A list of variables must be configured.

Sample for the drive (like U:) in the variable %drive%

```
[Settings]
```

```
UsbdmVariablesToOpenEnvironment=%drive%
```

Both mechanisms described below can be executed on user logon to deal with drives present at startup.

```
[Settings]
```

```
AutoRunOnLogon=1
```

Under Windows 2000 this does not work because there is no logon notification. We have to do it once on startup of the USB DLM service here:

```
[Settings]
```

```
AutoRunOnStartup=1
```

1. autorun.inf on the attached drive

The autorun.inf is a Windows mechanism. One function is the open= line to execute a command-line when

a drive is attached. With each Windows version this has been limited more and more. Under XP this works with CD-ROM drives only without further user inquiry. This is the reason for such great solutions as U3 drives or self-installer devices which are using a fake CD-ROM drive. Since Vista even this works after asking the user only.

USB DLM can execute the open= line in an autorun.inf on removable drives, hard drives, CD/DVD and Floppies drives.

Sample for drives with removable media:

```
[Settings]
AutoRunInf=1
```

Values for other types, add if required:

```
Removable: 1
Hard drives: 2
CD-ROMs 4
Floppies 8
Unknown 16
Remote 32
```

Sample for removable drives, hard drives and CD-ROMs:

```
[Settings]
AutoRunInf=7
```

Floppy drives have no insert notification, therefore AutoRun works (if at all) only when an external Floppy with a disk present is attached to the system.

Sample for opening a Windows Explorer window thru an autorun.inf on the attached drive or inserted media:

```
[Autorun]
open="%windir%\explorer" .
openstyle=max
```

On x64 systems the section [Autorun.Amd64] is read first. If the value is not found in this section then [Autorun] is read.

A window style can be suggested using an openstyle= line, this is USB DLM specific and not supported by Windows' autorun.inf.

Many programs consider this (as the Windows Notepad), other do not (as the Windows Calculator).

The available styles are:

```
max    maximized
min    minimized
```

hidden hidden

noactivate normal, but the window is not activated, it does not get the focus

Since Vista it's quite tricky to start a program from a service and get its window activated. If this is not required at all then please configure

openstyle=noactivate

If required a delay until the program is executed can be added. The default unit is milliseconds, 's' for Seconds and 'h' for hours can be used. The maximum value is 24 hours:

delay=10s

Of course the autorun.inf can be easily abused. Therefore USB DLM can protect this by a key:

```
[Settings]
```

```
AutoRunKey=MySecretKey
```

Only if the same line is found in the autorun.inf's [open] section on the drive the open= line is executed.

Another security option is to execute the AutoRun without admin privileges:

```
[Settings]
```

```
AutorunInfRestricted=1
```

If the current user isn't an admin, then this setting makes no difference.

Of course a SETUP.EXE started from a CD drive may then not have enough privileges to install its software...

Since V4.3 most extension as open1 to open9, wait, delay etc work here too.

Label and Icon from autorun.inf

Since V4.4 the items label and icon can be read from an autorun.inf file and written to the registry values DefaultLabel and DefaultIcon for this drive.

This can be useful if the autorun.inf is completely disabled or if it just does not work...

It must be activated separately for label and icon same way as show above.

Sample for Label and Icon on CDROM drives:

```
[Settings]
```

```
AutoRunInfLabel=4
```

```
AutoRunInfIcon=4
```

2. Global AutoRun settings in the USBDLM.INI

2.1 Triggered by volumes

```
[OnArrival]
open=%windir%\System32\calc.exe
```

This would start the Windows Calculator when a USB drive is attached or a media is inserted into a REMOVABLE or FIXED type drive.

If you need this for other types of drives, configure one or multiple DriveTypes. If you do so then the default "REMOVABLE or FIXED" is no more:

```
; for CD and Network drives
[OnArrival]
DriveType1=CDROM
DriveType2=REMOTE
open=%windir%\System32\calc.exe
```

The command can be executed without admin privileges (no difference if the active user is no admin):

```
[OnArrival]
open=%windir%\System32\calc.exe
restricted=1
```

or with full system privileges in the context "LocalSystem":

```
[OnArrival]
open=%windir%\System32\calc.exe
system=1
```

You will never see this program because it's in the "LocalSystem" context it's not allowed offhand to show a window on the user's desktop.

Using this is useful when system tools shall be started but the user is a restricted one as shown below in Sample 6.

Since Vista admin user work with a split access token. One incarnation is "limited" the other is not limited and called "elevated". This is what you get when processes started manually selecting "Run as administrator".

By default USBDLM starts processes limited. If you need something "elevated":

```
[OnArrival]
open=%windir%\System32\calc.exe
elevated=1
```

As parameter for the program you can use [variables](#) as %drive% for the drive (as X:) or %root% for its root folder (as X:\).

This can be made depending on criteria as described under [Drive letters depending on certain criteria](#). An additional criteria is the volume's drive letter, so a line as Letter=X is a criteria here! Only a single letter or an NTFS mountpoint work here in one Letter= line. An NTFS mountpoint needs a trailing backslash here to work.

Furthermore there is the default criteria MinDriveSize=1, so OnArrival events are executed by default only when the drive size is at least one Byte (a media is present in the drive). To execute OnArrival for drives without a media, configure MinDriveSize=- or MinDriveSize=0

USB DLM checks sections [OnArrival1] to [OnArrival99], and finally [OnArrival] (without a number).

A window style can be suggested using an openstyle= line as shown above.

To prevent the user skips the AutoRun by holding down the shift key you can set Force=1 in a section.

To prevent a process is started even it is already running, configure OneInstance=1:

```
[OnArrival]
open=%windir%\System32\calc.exe
OneInstance=1
```

To ensure the integrity of the executable file, an MD5 hash value can be configured:

```
[OnArrival]
open=%windir%\System32\calc.exe
md5=DDCD9FCD B7E1956E E69F8E58 B8C8BF0D
```

Only if the MD5 hash value of the executable is equal to the configured one, it is executed. The MD5 is case insensitive, spaces don't matter.

[MD5-File.zip](#)

You can also grab the open line from an autorun.inf file:

```
[OnArrival]
open=%drive%\autorun.inf
```

Sample 1:

- FotoSoftware for drives with a volume label "CANON_DC" or "NIKON_DC"
- nothing for drive X
- otherwise a maximized Explorer window, but not if it's a card reader without a card (size 0)

```
[OnArrival]
VolumeLabel1=CANON_DC
VolumeLabel2=NIKON_DC
open="C:\Program Files\PhotoSoftware\PhotoSoftware.exe" %root%
```

```
[OnArrival]
```


Letter=X
open=

;all others
[OnArrival]
open="%windir%\explorer.exe" %root%
openstyle=max

Sample 2:

- If the file DATA.TXT exist, copy it from the drive to C:\Data

[OnArrival1]
FileExists=%drive%\DATA.TXT
open="%windir%\System32\cmd.exe" /c copy "%drive%\DATA.TXT" "C:\Data"

cmd is the Windows command processor, /c means "execute command and end then", copy is a command which the cmd knows and copies files.

The same hidden:

[OnArrival]
FileExists=%drive%\DATA.TXT
open="%windir%\System32\cmd.exe" /c copy "%drive%\DATA.TXT" "C:\Data"
openstyle=hidden

Sample 3:

AutoMount a TrueCrypt container file with name secret.tc on T:, open an Explorer windows with the mounted TrueCrypt volume

[OnArrival]
FileExists=%drive%\secret.tc
open="C:\Program Files\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\secret.tc" /L T

[OnArrival]
DeviceType=TrueCryptVolume
open="%windir%\explorer" %root%

Sample 4:

AutoMount a TrueCrypt volume, remove its drive letter and open an Explorer window with the mounted TrueCrypt volume

;remove the volume's drive letter, it's useless
[DriveLetters]
DeviceType1=TrueCrypt
DeviceType2=ReadSharingViolation
Letter=-

;mount it on J:

```
[OnArrival]
DeviceType=TrueCrypt
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v %PartitionName% /L J
```

;an Explorer window with the mounted TrueCrypt volume

```
[OnArrival]
DeviceType=TrueCryptVolume
open="%windir%\explorer" %root%
```

%PartitionName% is a USB DLM variable {linkID=} which USB DLM replaces with something like \Device\Harddisk2\Partition1 as expected by TrueCrypt.

The DeviceType ReadSharingViolation applies for drives already mounted by TrueCrypt. Since TrueCrypt get exclusive access USB DLM cannot read test data - the attempt it's rejected with ERROR_SHARING_VIOLATION, that's what the name comes from.

This usually happens when the USB DLM service is manually restarted. We want the letter removed but the volume shall not being mounted again by TrueCrypt, that's what this sample does.

It does not work if a "fixed" drive is completely used as TrueCrypt container because in this case there is no logical drive on the disk and USB DLM isn't triggered. Create a single partition then and use this a TrueCrypt container. This is suggested by TrueCrypt and prevents the Windows Disk Management from screwing up the volume by "initializing" the drive which sounds less harmful than the Windows Explorer's suggestion to format the TrueCrypt container volume...

Sample 5:

- a USB drive on U: and create a share Drive_U for it, on "safe removal" delete the share (openstyle=hidden prevents the console window from appearing when starting the NET.EXE):

```
[DriveLetters]
Letter1=U
```

```
[OnArrival]
Letter=U
open="%windir%\System32\net.exe" share FlashDrive_U=U:\
openstyle=hidden
```

; on safe removal the share must be deleted, otherwise the safe removal would be denied if the share is accessed

```
[OnRemovalRequest]
Letter=U
open="%windir%\System32\net.exe" share FlashDrive_U /D
openstyle=hidden
```

; in case there was not safe removal

```
[OnRemoval]
Letter=U
open="%windir%\System32\net.exe" share FlashDrive_U /D
openstyle=hidden
```

Since Vista with active UAC a system=1 or elevated=1 is required to get the privileges required to create a share.

Sample 6:

- a USB flash drive at A:, if A: is in use, remount A: to B: and assign A: then:

```
;mount to A: if available, otherwise unmount
[DriveLetters]
Letter=A
Letter=-

;if unmounted, remount A: to B: and assign A:
[OnArrival]
Letter=-
open=c:\A_to_B_and_drive_to_A.cmd %VolumeName%
system=1
```

In the command script c:\A_to_B_and_drive_to_A.cmd:

```
ReMount A: B:
mountvol A: %1
```

%VolumeName% is a USB DLM variable {linkID=} which USB DLM replaces by something like \\?\Volume{aa6d706a-15da-11dc-a38f-0013d31dd4c5}\, as expected by the Windows command-line tool MountVol.

ReMount is my command-line tool for changing drive letters. Download:
http://www.uwe-sieber.de/drivetools_e.html#remount

Assign A: to the previous drive when the USB flash drive has been removed:

```
[OnRemoval]
Letter=A
open=ReMount B: A:
system=1
```

The line system=1 makes USB DLM executing the command line in the context "LocalSystem" where itself is running. This is required for restricted users when tools as MountVol or ReMount are executed, because they will not work without admin privileges. And they don't pop up... Furthermore such an item is executed too when no user is logged on.

Multiple open commands

Since V4.3 you can let execute up to 10 commands. Use open1= to open9= and open= then. Additional parameters as openstyle, restricted etc must be numbered then too and have effect on the open line with the same number, so in contrast to all other section types there is a relation between items with the same number!

Sample 7 (USB DLM V4.3+):

- start the Windows Calculator twice, one restricted and one normal

```
[OnArrival]
open1=calc.exe
restricted1=1
open2=calc.exe
restricted2=0
```

- start the Windows Calculator and the Windows Notepad (maximized) as soon as the Calculator ends or after 10 Seconds wait

```
[OnArrival]
open1=calc.exe
wait1=10000
open2=notepad.exe
openstyle2=max
```

Since V5 the numbering isn't required, an open line starts a new parameter set:

Sample 7 (USB DLM V5.0):

```
[OnArrival]
open=calc.exe
wait=10000

open=notepad.exe
openstyle=max
```

Sample 8:

- remove the drive letter of a USB drive, dismount its file system by means of [EjectMedia](#), copy an image to it by means of [DD für Windows](#), finally safely remove it by means of [RemoveDrive](#):

```
[DriveLetters]
Letter=-
```

```
[OnArrival]
Letter=-
open="C:\Tools\EjectMedia.exe" %VolumeName% -D
wait=1000
```

```
open="C:\Tools\DD.exe" -if=C:\usb-image1.bin -of=\\.\%PureVolumeName%
wait=600000
```

```
open="C:\Tools\RemoveDrive.exe" %VolumeName% -L
```

For direct write to a volume or a disk device admin privileges are required. Since Vista with active UAC you need a elevated2=1 or system2=1 too, with the latter you will not see the console windows because it runs in the "Local System" context.

If you want to write to the disk device then the target for DD is \\.\%PhysicalDrive%DeviceNumber%

2.2 Triggered by disks

Especially for disks without partitions there are the events [OnDiskArrival], [OnDiskRemovalRequest], [OnDiskRemoval] which work like the volume triggered events described above. Of course volume specific criteria cannot work here.

Sample:

; mount a disk which is entirely used as TrueCrypt container to J:

```
[OnDiskArrival]
```

```
DeviceType=TrueCrypt
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v \Device\Harddisk%DeviceNumber%\Partition0  
/letter J
```

AutoRuns on user logon / user switch

Configured AutoRun events are executed by default on arrival of a drive or on insertion of a media only. If USBDLM shall do this on user logon or user switch:

```
[Settings]
```

```
AutoRunOnLogon=1
```

With an auto logon the USBDLM service may not be started yet on logon, so it misses the logon event. Workaround: When there are fewer than two minutes since the system is started and the a user is already logged on when the USBDLM services starts, then the AutoRuns are executed too.

Windows 2000 does not support the required notification messages, so USBDLM cannot AutoRun on logon.

Diagnostic

If there are problems with the command line parameters, working directory, being admin or not, window style etc., it might be helpful to check this by executing my debug tool TestCommandLine:

<http://www.uwe-sieber.de/files/testcommandline.zip>

By default it closes after 10 Seconds. By clicking on the countdown the timeout starts over again. By command-line -t:xx a different timeout in seconds can be specified.

The window of TestCommandline does not become active and it ignores the given window style (but shows it).

If a program is started hidden or in the context "LocalSystem" then it is invisible. In these cases USBDLM since V4.7.2 redirects the output of console programs to the logfile. By default with log-level 3, another log-level can be configured, e.g.

[Settings]

RedirectLogLevel=2

AutoRun on and after Removal

In analogy to the OnArrival function USBDLM can execute a command-line when a drive is "prepared for safe removal" and after a drive has been removed.

1. On preparation for safe removal

When a USB or Firewire drive becomes "prepared for safe removal" the USBDLM can react while the drive is still available.

This should not take too long, the maximum time is 30 Seconds under XP, and 15 Seconds since Vista/Win7. But while the notification is processed, no other events can be handled. Therefore USBDLM wait up to 10 Seconds only. If the started process is still running after this time, then USBDLM rejects the removal request. Windows then says "USBDLM prevents the removal...".

The OnRemovalRequest section for a drive should be created before the drive in question is attached.

As with OnArrival, the parameter Letter is a criterion here.

Sample to copy the file c:\test.txt to the folder \backup on the drive to remove:

```
[OnRemovalRequest]
open="%windir%\System32\cmd.exe" /c copy "C:\test.txt" %drive%\backup
```

Same, but for drive X: only:

```
[OnRemovalRequest]
Letter=X:
open="%windir%\System32\cmd.exe" /c copy "C:\test.txt" %drive%\backup
```

As in the first sample but only if the file \backup\test.txt exists on the drive to remove:

```
[OnRemovalRequest]
FileExists=%drive%\backup\test.txt
open="%windir%\System32\cmd.exe" /c copy "C:\test.txt" %drive%\backup
```

Unmount a TrueCrypt volume T: when the file secret.tc is found on the drive to remove:

```
[OnRemovalRequest]
FileExists=%drive%\secret.tc
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /d T
```

This works too if the drive letter of the container drive is removed. %drive% is replaced then by the volume name which works too.

Unfortunately TrueCrypt needs its drive letter for unmounting. It cannot unmount a container by its name even this should be no problem. Ask the TrueCrypt guys...

For complex operations it's better to do the job in a batch file and finally prepare the drive for safe removal by means of my command-line tool RemoveDrive:

<http://www.uwe-sieber.de/files/removedrive.zip>

Sample to kill a process running on the drive to remove and wait a bit:

```
[OnRemovalRequest]
Letter=X
open=tskill MyProcess
afterdelay=2000
```

Killing a process is bad. The more friendly way is to send a WM_CLOSE message to the application window. USBDLM can find the window depending on its window text and (optionally) its class name. In case the application needs some additional time to completely close after the main windows is gone, an "after delay" of up to 5 Seconds can be specified.

Sample to find and close the Windows Calculator (the calc.exe) and wait 2 Seconds after sending WM_CLOSE:

```
[OnRemovalRequest]
Letter=X
CloseWindowText=Calc
CloseWindowClass=SciCalc
afterdelay=2000
```

[Wildcards](#) can be used when the windows text depends on the opened document:

```
[OnRemovalRequest]
Letter=X
CloseWindowText=* - Notepad
CloseWindowClass=Notepad
afterdelay=2000
```

The window class can be determined by means of my tool WinInfo:

<http://www.uwe-sieber.de/files/wininfo.zip>

2. On fail of the safe removal

When the drive to prepare for safe removal is still accessed then the safe removal fails. For instance a just deleted share can be restored then:


```
[OnRemovalFailed]
```

```
Letter=U
```

```
open=net share FlashDrive_U=U:\
```

3. After removal

Sample: Drive A: has been removed, another drive shall be mounted to A:

```
[OnRemoval]
```

```
Letter=A
```

```
open=mountvol A: \\?\Volume{433619ea-c6ea-11d9-a3b2-806d6172696f}\
```

```
system=1
```

MountVol is a command-line tool which comes with Windows 2000 and higher.

system=1 makes USBDLM execute it in the System context to give it the required privileges in case the user isn't an admin. It's invisible then.

Furthermore, since V5.2.5 there are OnSafeRemoval and OnUnsafeRemoval, but only for real volumes and not for virtual ones as TrueCrypt volumes. Sample:

```
[OnUnsafeRemoval]
```

```
open="%WinDir%\System32\mplay32.exe" /play /close "%WinDir%\Media\chord.wav"
```

```
windowstyle=hidden
```

mplay32.exe was available up to XP only, since Windows 7 you need a 3rd party tool or the powershell to play a WAV without ending up with a open player.

Or use [USBDLM_usr.exe](#) to show a messagebox:

```
[OnUnsafeRemoval]
```

```
open="%USBDLMPATH%\USBDLM_usr.exe" -msgbox -type:0 -icon:0x30 -title:Warning -  
text:%BusType% device %FriendlyName% was unsafely removed. This is not safe!
```

Limitation

All these removal events are triggered only if the volume was present for at least 5 Seconds.

AutoRun for Disk Devices

In analogy to the AutoRun function for storage volumes, USB DLM can execute a command-line on arrival and removal of a disk device, which is handy if the disk has not partitions, e.g. if it is entirely used as a TrueCrypt container.

As criteria all disk specific ones can be used, as the disk's device ID, MinDriveSize, MaxDriveSize, PortName.

All non-drive-specific criteria are available here too. These are [Files](#), [User](#), [computer name](#), [running process](#), [operating system](#) and [time](#).

Sample for mounting a TrueCrypt container:

```
;mount it on J:  
[OnDiskArrival]  
DeviceType=TrueCrypt  
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v \Device\Harddisk%DeviceNumber%\Partition0  
/letter J
```

```
;dismount J: on safe removal of the disk  
[OnDiskRemovalRequest]  
DeviceTypes=TrueCrypt,ReadSharingViolation  
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dJ
```

The other parameters known from [\[OnArrival\]](#), as openstyle, system etc work here too.

AutoRun for other devices

In analogy to the AutoRun function for storage volumes, USBDLM can execute a command-line on arrival and removal of any device.

The one and only device specific criteria is the device ID.

You find the device ID in the Windows Device Manager. Right-click the device -> Properties. Select the "Details" tab. Here the "Device Instance ID" is already selected. Click on the ID and press "Ctrl+C" to copy the ID into the Windows Clipboard.

XP before SP2 show the tab "Details" when the environment variable DEVMGR_SHOW_DETAILS is set to 1. This REG file adds this variable:

[devmgr_show_1.reg](#)

It takes effect after the next logon.

Since V4.7.1 all criteria not related to a newly arrive drive are available here too. These are [Files](#), [User](#), [computer name](#), [running process](#), [operating system](#) and [time](#).

Sample for starting the calculator on arrival of a certain USB device:

```
[OnDeviceArrival]
DeviceID1=USB\VID_090C&PID_1000\A74000000000097
open=calc
```

The same after removal:

```
[OnDeviceRemoval]
DeviceID1=USB\VID_090C&PID_1000\A74000000000097
open=calc
```

Hint: There is no [OnDeviceRemovalRequest].

The other parameters known from [\[OnArrival\]](#), as openstyle, system etc work here too.

AutoRun on System Events

In analogy to the AutoRun function, USBDLM can execute a command-line on system events.

The events are

OnServiceStart	Start of the USBDLM service, before checking the drive letters
OnServiceStarted	Start of the USBDLM service, after checking the drive letters
OnServiceStop	Stop of the USBDLM service
OnServiceShutdown	Shutdown of the USBDLM service (on Windows shutdown)
OnUserLogon	After user logon
OnUserLogoff	After user logoff
OnUserLock	After locking the workstation
OnUserUnlock	After unlocking the workstation
OnUserConnect	"connecting" of mouse and keyboard with the desktop (e.g. after logon or unlocking the workstation)
OnUserDisconnect	"disconnecting" of mouse and keyboard from the desktop (e.g. after logoff or locking the workstation)
OnUserActivated	After user logon, after unlocking the workstation and after switching to a user
OnUserDeactivated	After logoff, after locking the workstation and after initiating a user switch
OnSleepRequest	On standby and hibernation; Windows does not wait for more than 2 seconds No more available since Vista/Win7
OnSleep	On standby and hibernation; in real live it might be triggered on wakeup
OnResume	On wakeup from standby and hibernation
OnUndockRequest	On soft "Undock" (e.g. thru the start menu)
OnUndock	After successfully soft "Undock" (e.g. thru the start menu)
OnDock	After "Docking" but only if the "Undock" was done thru the start menu

OnUserConnect and OnUserDisconnect mean the connection of the input devices (mouse and keyboard) with the desktop. Under XP OnUserDisconnect is not triggered before a different user logs on, so it is not useful.

Under Vista and Windows 7 OnUserLock and OnUserUnlock are not triggered when switched from one user to another. Under XP they are triggered but not in the following situation: User1 is logged on, switches to the logon screen, User2 logs on, User2 logs off, User1 logs on again. Even User1 got a Lock before, XP does not trigger an Unlock.

Because neither Connect/Disconnect nor Lock/Unlock are reliable under XP, USBDLM generates OnUserActivated and OnUserDeactivated which are reliable.

For events where the user is not logged on a system=1 is required to execute the program in the "Local System" context.

OnSleepRequest and OnSleep does not guarantee that something is executed (and finished) before the system goes to sleep. OnSleep is often triggered in fact on wake up...

OnSleepRequest does not exist under Vista, Windows 7 and Server 2008.

If the Windows option "Ask for password on resume from standby" is activated then under XP the workstation gets locked on wakeup, so then events OnUserLock, OnUserDisconnect and OnUserDeactivated are triggered only then. USB DLM can lock the workstation on standby, so the event are triggered before going to sleep, but this is not reliable. Sometimes it does not work or the system stops going to sleep.

[Settings]

LockWorkstationOnSleep=1

The other parameters known from [\[OnArrival\]](#), as openstyle, system etc work here too. Furthermore multiple open lines can be defined by numbering them.

For events which are triggered when the user is already gone the program is executed in the System context.

Sample for creating a network drive on user logon:

[OnUserLogon]

open="%windir%\System32\cmd.exe" /c net user X: \\server\share /user:username password
openstyle=min

Since V4.5 multiple sections of a type can be created but the only criteria available are the [user criteria](#) and the [file exists criteria](#).

The user criteria makes no sense in a OnUserLogoff section because it's information is already gone then.

Sample for starting the WIA service for admins and stop it for non-admins on logon:

[OnUserLogon]

IsUserAdmin=1

open=net start stisvc
system=1

[OnUserLogon]

IsUserAdmin=0

open=net stop stisvc
system=1

system=1 makes the NET.EXE being started in the "System" context, this ensures the console window is not seen by the user and the required privileges are available.

Balloon Tips

USB DLM can show a balloon tip with information about the assigned mountpoints of newly attached drives.

```
[BalloonTips]
Enabled=1
Timeout=6000
```

Timeout is the duration the balloon tip is shown in milliseconds.
5 seconds (5000 ms) plus an additional half second per mount point are default.

The timeout starts again when the mouse is moved over the balloon tip.

Since V5.0 USB DLM can show balloon tips on removal:

```
[BalloonTips]
Enabled=1
EnabledOnRemoval=1
Timeout=6000
TimeoutOnRemoval=4000
```

Default for EnabledOnRemoval is the value of Enabled, default timeout is 2/3 of the Timeout value.

USB DLM includes text resources for some languages. You can change some or all text snippets in the USB DLM.INI:

Sample for the build in English texts:

```
[BalloonTips]
Text_drives=drives
Text_partitions=partitions
Text_media=media
Text_cddvd=CD/DVD
Text_mounted_to=mounted to
Text_and=and
Text_safely_removed=Safely removed
Text_unsafely_removed=Unsafely removed
Text_removed=Removed
Text_ejected=Ejected
Text_NetworkDrive=Network Drive
Text_SubstDrive=Subst Drive
```

For Text_safely_removed and after I would like to add text resources for French, Italian, Spanish and

Portuguese as USB DLM V4 already has for the first six...
mail@uwe-sieber.de

Resulting sample texts are

- for attached drives with one volume, as flash drives or harddrives with one single partition:
mounted to P:\
- for disks with multiple partitions:
2 partitions mounted to P:\ and O:\
- for devices with multiple drives, as multislot cardreaders
3 drives mounted to J:\, P:\ and O:\
- for an inserted media, as a flash card (only with NoMediaNoLetter)
media mounted to P:\
- for an inserted CD, DVD, BD (only with NoMediaNoLetter)
CD/DVD mounted to P:\

If in your language "mounted to" differs in singular and plural, then you can use the additional item
Text_mounted_to_plr.

In Slavonic languages the plural for 5 or more is different from 2..4. Here we can configure Text_drives_5,
Text_partitions_5 and Text_mounted_to_5, which are used when 5 or more drives or partitions are to show.

Many different variants of putting together the texts are considered. Therefore a fully free configurable
balloon text is too much effort.

But for a simple single drive mounted to a drive letter it's no problem.

Sample for a balloon tip with drive letter, volume label, file system and USB speed:

```
Text_singledrive=mounted to %Root%  "%VolumeLabel%" [%FsName%] - %UsbSpeed%
```

Results in:

```
mounted to X:\  "Backup" [NTFS] - SuperSpeed
```

%Root%, "%VolumeLabel%" and %FsName% are USB DLM variables [{linkID=}](#)

To get "%VolumeLabel%" [%FsName%] in the upper sample right aligned, a "center" can be defined. The
placeholder for this is " !! ". It is replaced first by six space chars. Is there more space because of a long
balloon title, then more spaces are added.

To get some distance to the right border, adding spaces does not work because they are removed when the
INI is read in. Use underscores instead, they are replaced by spaces when used at the end.

Sample:

```
Text_singledrive=mounted to %Root% !!  "%VolumeLabel%" [%FsName%]_____
```

For inserted CD/DVD there is Text_singleCdDvd and for other removable media Text_singleMedia (both
only when a mountpoint is assigned on insertion).

Since V5.0 the Balloontips can be customized way more flexible, see [Balloontips Customized {linkID=9}](#).

No Balloontips for certain drives

More and more often we come across USB devices as printers, web cams or WLAN sticks which have a "SelfInstaller". It's a fake CD-ROM drive which is used to install their drivers by means of the Window AutoPlay facility for CD-ROM drives.

USB DLM shows a BalloonTip for these CD-ROMs. To prevent this, the BalloonTip can be disabled for certain drives by their hardware ID or their "friendly name". Sample:

```
[BalloonTipsExclude]
DeviceID=USB\VID_057C&PID_62FF
DeviceID=selfinstall
```

The ID has to be eight or more characters or using [Wildcards {linkID=115}](#). You can use UsbDriveInfo to get the string, see [Letters by DeviceID {linkID=39}](#)

Actions on click on the Balloontip

Similar to autorun events you can define actions on left, right and middle click on the balloon.

```
;on left click, open a simple Explorer window with the drive
[OnBalloonClick]
open="%windir%\explorer" %drive%
```

```
;on right click, open a foto software
[OnBalloonRClick]
open="C:\Program Files\FotoSoft\fotosoft.exe" %drive%
```

You can define several events depending on criterions as shown for [AutoRun {linkID=90}](#).

When more than one drive is shown in the balloon, then the event works for the first one which has a drive letter only.

Under XP explorer /e shows an Explorer windows with a drive tree. Don't use this here because XP needs about 4 seconds to put a new drive into the tree. If you use it before then it may create an empty item in the tree for the new drive...

Several more [settings for balloon tips {linkID=8}](#) can be made.

Balloon Tips Settings

Suppress Windows Balloons

While an external drive is attached for the first time and after safe removal Windows shows more or less useful balloon tips. USBDLM can suppress these balloons if required to show its own. Default is 1, see below.

Under Windows 8 the Windows balloons are at a different place (and called "toast notifications"), USBDLM does not care about these. Under Windows 10 the toast notifications are in the classic balloon tip place and more helpful, but USBDLM still can do better...

```
[BalloonTips]
SuppressWindowsBalloons=1
```

- 0: Off
- 1: Close on appearance
- 2: Temporarily disable Windows balloon-tips

The Windows "Safely remove hardware" tray icon does not like disabled Windows balloon-tips, it stops work then, so SuppressWindowsBalloons=2 is not good if you need it.

Windows 10 Style

Since Windows 10 the Windows balloon tips look different, USBDLM tries to emulate this. Under Windows 8 only balloon tips for new drives had the new design while all others had the classic style as under Windows 7.

By default USBDLM emulates the Windows 10 style if running on Windows 10. If you don't like it:

```
[BalloonTips]
Win10Style=0
```

The colors for the Windows 10 style are derived from the Windows accent color found in the registry here: HKCU\Software\Microsoft\Windows\DWM\AccentColor

If you know better you can set the colors like so:

```
[BalloonTips]
BackColor=0xA02020
CaptionColor=0xFFFFFFFF
TextColor=0xFFCFCF
```

Compact Style for Multislot Cardreaders

With NoMediaNoLetter=1 a BalloonTip for a four-slot-reader may look like this:

```
MultiReader USB device  
mounted to ---, ---, --- and ---
```

On first view useless it informs that a card reader has four drives and there is no drive letter assigned. Since V5.0 you can turn it into a compact version if there is none or only one slot with a media present:

```
[BalloonTips]  
MscrCompact=1
```

Multiple mountpoints

Since V4.3 USBDM can deal with multiple mountpoints. By default the balloon tips show up to two mountpoints per volume. If you need more:

```
[BalloonTips]  
MaxMountPointsToShow=3
```

Or if you want to see the first only:

```
[BalloonTips]  
MaxMountPointsToShow=1
```

By default a drive letter is shown first, then the NTFS mountpoint(s). If you want the reverse order:

```
[BalloonTips]  
MountPointsReverse=1
```

Show the drive size

```
[BalloonTips]  
ShowSize=1
```

This way the drive size is shown behind the device name, like this:

```
Corsair Flash Voyager USB device - 8 GB
```

Default is on since V5.0.

Round the drive size

```
[BalloonTips]  
RoundSize=1
```

Makes USBDM show fixed sizes as 256 MB or 512 MB instead of values like 262 MB or 508 MB and furthermore full Gigabytes.

Default is on since V5.0.

Show the USB connection speed

Since V5.0 USBDLM shows in the BalloonTip icon for USB drives their connection speed - e.g. an HS or SS for High-Speed or SuperSpeed. If you don't want this:

```
[BalloonTips]  
UsbSpeedIcon=0
```

Show maximum supported USB speed of the Device in the balloon's title

Since 5.0 USBDLM shows the device's USB version in the balloon title, e.g. "USB 3.0 Device" instead of "USB Device".

Alternatively it can show the maximum speed as text, e.g. "USB SuperSpeed Device":

```
[BalloonTips]  
UsbSpeedTitle=2
```

0 -> off

1 -> max Speed as Text

2 -> USB version as number (Default)

SuperSpeedPlus is probably detected under Windows 10 only.

Wait for the last volume

When a device with multiple drives is attached as a USB hard drive with multiple partitions or and USB multislot cardreader, USBDLM can either wait until all volumes are processed or update the balloon tip for each volume. Default is off.

```
[BalloonTips]  
WaitForLastVolume=0
```

Balloontips for TrueCrypt volumes

Since V4.3 USBDLM shows balloon tips for mounted TrueCrypt volumes. If it shall not do so:

```
[BalloonTips]  
ShowTrueCrypt=0
```

Balloontips for Network and Subst drives

Since V4.6 USBDLM can show balloon tips for newly created Network and Subst drives. In fact we talk about any non physical drive here which has a drive letter but no real storage volume. USBDLM calls them

"virtual" drives.

```
[BalloonTips]
ShowVirtual=1
```

Default is on since V5.0.

Since Vista with the User Account Control active, the user is split into two incarnations, one restricted and one elevated. By default USBDLM sees drives of the restricted one only, if you need virtual drives of the elevated one too, configure 2 instead 1. This makes the USBDLM_usr.exe loaded twice, one in each context, to receive notifications from both.

In the balloon tips USBDLM then shows an additional "(elevated)" if the drive came for the elevated account and "(normal+elev)" if for both (e.g. for TrueCrypt volumes or for network drives if [EnableLinkedConnections](#) is active.

The texts can be customized (also set empty):

```
[BalloonTips]
Text_ContextNormal=normal
Text_ContextElevated=elevated
Text_ContextBoth=normal+elevated
```

Balloontips for U3 drives

By default USBDLM balloons show the data part of an U3 drive only. If you want to see both, data and system (CDROM) drive:

```
[BalloonTips]
ShowU3CDROM=1
```

The data part is shown first.

Codepage

Since V4.6 USBDLM works with Unicode (UTF-16) - no more codepages and charsets.

If the USBDLM.INI file is an ANSI file it must be converted to Unicode when USBDLM reads it. By default the Windows default codepage is used for the conversion. If a different one is required, it can be defined:

```
[BalloonTips]
Text_codepage=1251
```

If the USBDLM.INI file is a Unicode file then this has no effect. UTF-8 and UTF-16-LE (Notepad calls it Unicode) can be read.

Right to left mode text as used for Arabic and Hebrew isn't supported yet.

Balloon Tips Customized

Since USBDLM V5.0 the balloon tips of drives with a single volume can be customized by means of BalloonTipsOnXXX sections which work identical to DriveLetters or OnArrival sections.

Here the supported sections:

- [BalloonTipsOnArrival] Volume arrived
- [BalloonTipsOnRemoval] Volume removed (safe or unsafe is unknown, e.g. Virtual Volumes)
- [BalloonTipsOnSafeRemoval] Volume safely removed
- [BalloonTipsOnUnsafeRemoval] Volume unsafely (surprisingly) removed
- [BalloonTipsOnMediaArrival] Media inserted (mountpoint was already assigned)
- [BalloonTipsOnSafeMediaRemoval] Media removed or ejected and was dismounted before
- [BalloonTipsOnUnsafeMediaRemoval] Media removed or ejected and was mounted before

Title defines the bold printed title of the balloontip, Text the text, in both ~~USBDLM-Variable~~[{linkID=}](#) can be used.

Icon defines either a USBDLM ~~internal icon number~~[{linkID=130}](#), an ICO file or a icon resource in an EXE or DLL file. Even the notation looks like the one used by the Windows Explorer the icon index works different here (without groups). It is what the tool [ResHacker](#) shows.

Samples

On Arrival

```
; TrueCrypt volumes with size, name of the file system and the blue key icon from the TrueCrypt.exe
```

```
[BalloonTipsOnArrival]
```

```
DeviceType=TrueCryptVolume
```

```
Title=TrueCrypt Volume - %VolumeSize%
```

```
Text=mounted to %Root% ( %FsName% )
```

```
Icon="%ProgramFiles%\TrueCrypt\TrueCrypt.exe",2
```

```
; USB drives with device name, maximum USB speed, actual connection speed and drive size
```

```
; (BusType=USB is default in these sections too)
```

```
[BalloonTipsOnArrival]
```

```
Title=%FriendlyName% %BusType% %UsbVersion% %UsbMaxSpeed% Device at %UsbSpeed% - %DriveSize%
```

```
Text=mounted to %Root% ( %FsName% )
```

On Removal

; TrueCrypt volumes with size, name of the file system and the orange key icon from the TrueCrypt.exe
[BalloonTipsOn**Removal**]
DeviceType=TrueCryptVolume
Title=%FriendlyName% - %VolumeSize%
Text=removed: %Root% (%FsName%)
Icon="%ProgramFiles%\TrueCrypt\TrueCrypt.exe",6

On safe removal

; all BusTypes
[BalloonTipsOn**SafeRemoval**]
BusType=ANY
Title=%FriendlyName% %BusType% Device - %UsbSpeed% - %DriveSize%
Text=safely removed from %Root% !! %FsName%

On unsafe removal

; all BusTypes
[BalloonTipsOn**UnsafeRemoval**]
BusType=ANY
Title=%FriendlyName% %BusType% Device - %UsbSpeed% - %DriveSize%
Text=surprisingly removed from %Root% !! %FsName%

On insertion of a media

;USB drives, camera icon from the Windows7 shell32.dll
[BalloonTipsOn**MediaArrival**]
Title=%FriendlyName% %BusType% Device - %DriveSize%
Text=FlashCard inserted: %Root% (%FsName%)
Icon=%windir%\System32\shell32.dll,1347

On safe removal/ejection of a media

;USB
[BalloonTipsOn**SafeMediaRemoval**]
Title=%FriendlyName% %BusType% Device - %DriveSize%
Text=FlashCard safely removed: %Root% (%FsName%)

On unsafe removal/ejection of a media

;USB
[BalloonTipsOn**UnsafeMediaRemoval**]
Title=%FriendlyName% %BusType% Device - %DriveSize%
Text=FlashCard unsafely removed: %Root% (%FsName%)

Variables

Configuring AutoRun, Explorer Label and mounting into NTFS folders some variables can be used:

Variable	Description	Sample
%DriveLetter%	drive letter	X
%Drive%	drive	X:
%Root%	drive root	X:\
%BusType%	connection type	USB
%DevName%	device name	Corsair Flash Voyager
%FriendlyName%	device name	Corsair Flash Voyager
%DriveDeviceId%	drive's device id	USBSTOR\DISK&VEN_CORSAIR&PROD_FLASH...
%CtrlDeviceId%	controller' dev id	USB\VID_090C&PID_1000\...
%UsbSerial%	USB serial number	123456789 or ---
%UsbPortName%	USB port name	3-1-1
%UsbSpeed%	USB speed	High-Speed
%UsbMaxSpeed%	USB dev max speed	Super Speed
%UsbVersion%	USB-Version	2.0
%Label%	drive name	My flash drive
%VolumeLabel%	drive name	My flash drive
%LetterFromLabel%	letter from label	X if the label is Drive_X
%DriveType%	drive type	REMOVABLE
%FsName%	file system	FAT
%Size%	volume size	2 GB
%VolumeSize%	volume size	2 GB
%DiskSize%	drive size	20 GB
%DriveSize%	drive size	20 GB
%KernelName%	kernel name	\Device\Harddisk3\DP(1)0-0+d
%PartitionName%	Partition name	\Device\Harddisk2\Partition1
%DeviceNumber%	device number	2
%PartitionNumber%	partition number	1
%DiskSignature%	disk signature	9810ABEF
%VolumeName%	volume name	\\?\Volume{GUID}\
%PureVolumeName%	pure volume name	Volume{GUID}
%VolumeNameGuid%	volume name's GUID	{GUID}
%Target%	subst/net target	\\MyServer\public
%Date%	Date (yyymmdd)	090331
%Date4%	Date (yyyymmdd)	20120331
%DateISO%	Date (yyyy-mm-dd)	2013-03-31
%Time%	Time (hh:mm:ss)	12:00:00
%UserName%	user name	Administrator
%UsbdmPath%	USBDLM folder	C:\Tools\USBDLM

These variables can also be made available as environment variables for processes started by USBDLM. Which variables are required must be configured.

Sample:

[Settings]

UsbdmVariablesToOpenEnvironment=DevName, FsName, VolumeLabel

UsbDriveInfo can show the variables (Options -> Show USBDLM Variables).

Wildcards

Configuring the criterions DeviceID, VolumeLabel, VolumeSerial, UsbPortName, UserName and UserGroup wildcards can be used:

The asterisk '*' stands for null to n characters and the question mark '?' for exactly one.

It's all case insensitive.

Sample	fits for	does not fit
Backup?	Backup1, Backup2, BackupA, BackupB	Backup, Backup01
Backup*	Backup, Backup1, Backup01, Backup001	Back
*Backup	Backup, MyBackup	Backup1
ck	Backup, Block	Bak
B*up	Backup, Buckup, Bup	Backu
2-1-*	2-1-1, 2-1-7, 2-1-1-3	2-1, 1-1-1
2-1-?	2-1-1, 2-1-7	2-1, 2-1-1-3
USB\VID_090C*	USB\VID_090C&PID_1111.	USB\VID_2222...
admin*	Admin, Administrator	adam

Ignore a drive completely

There are some devices around which behaves strange. USBDLM can ignore such drives. The right drives are configured by means of their device ID, see [Letters by Device ID](#) how to get it.

Sample for the Apple iPod whose iTunes-Software deals with it's drive letter too:

```
[IgnoreDevices]
DeviceID1=ven_apple&prod_ipod
```

DeviceID1 to DeviceID9 can be configured.

Ignoring the Apple iPod is already build in because the iTunes Software seems to be triggered by the iPod's drive letter and when USBDLM changes it, iTunes gets confused.

If it shall be handled by USBDLM anyway then force it by using a criterion DeviceID in its drive letter section:

```
[DriveLetters]
DeviceID=ven_apple&prod_ipod
Letter=X
```

Windows Explorer Icon and Label

USBDM can put some values into the Windows registry to make the Windows Explorer use a special icon and label for a drive.

But the Explorer's behavior changed with each Windows version and even each XP service pack. Sometimes it works, sometimes it does not. It's not USBDM's fault.

At drive letters A: and B: XP seems to love its floppy icons and floppy names. Here is a high probability that XP ignores the default icon and label settings.

A default label is usually used only if the drive has no volume label. But sometimes it's used even the drive has a volume label. The Windows behavior seems to be unpredictable.

The configuration is similar to AutoRun. You can configure criterions, the first section with criterions which fit to the storage volume is used.

The registry settings are made with the flag "volatile", so Windows removes them on restart, USBDM recreates them if required.

USBDM deletes them on removal of the drive which they are made for.

Sample to get a camera icon for drives with volume label CANON_DC:

```
[Explorer]
VolumeLabel1=CANON_DC
DefaultIcon=shell32.dll,139
```

The icon is found in the Windows shell.dll of XP and higher at index 139 (counted from 0).

Or using an ICO file on the drive:

```
[Explorer]
FileExists=%drive%\usb.ico
DefaultIcon=%drive%\usb.ico
```

Sample to get a label "USB-Drive" for USB drives:

```
[Explorer]
BusType=USB
DefaultLabel=USB-Drive
```

You can use [variables](#) {linkID=} here.

Sample for the device name for USB drives which have no volume label:

```
[Explorer]
VolumeLabel=-
BusType=USB
DefaultLabel=%DevName%
```

If you want an empty DefaultLabel then configure DefaultLabel=-

In this case USBDLM writes a space as DefaultLabel because an empty value makes the Explorer show the default name as "Local drive".

Under Vista and Windows 7 with the fonts "Segoe UI" you can configure the Unicode character U+200B (zero width space). Of course the USBDLM.INI must be stored then in Unicode format.

The XP fonts Arial, Verdana etc do not support this character.

If you want no DefaultLabel then configure DefaultLabel=--

In this case USBDLM writes an empty string.

Define in DriveLetters Sections

Since V4.6 DefaultLabel and DefaultIcon can be defined in the volume's DriveLetters section.

If there is either DefaultLabel or DefaultIcon defined in the volume's DriveLetters section then USBDLM does not look for an Explorer section!

Sample:

```
[DriveLetters]
DeviceID=USB\VID_9876&PID_5432
Letter=X
DefaultLabel=%DevName%
```

Drives in "Safely Remove Hardware"

Most internal Cardreaders are USB devices and as such they are listed in the "Safely Remove Hardware" facility. USB devices have no chance to say "I'm an internal device and not removable", so Windows cannot distinguish between internal and external USB devices.

USB DLM can remove this removable flag from the registry. This hides a drive from the list.

Since **Windows 7** the Safely Remove Hardware facility can do both, the "classic" safe removal of the whole device and the ejection of a media. Therefore it will show here "Removable" drives even USB DLM made the registry changes. But at least the function to safely remove the whole device is deactivated, only the real media ejection stays available.

Whenever the driver of the device starts up the value is written again. Therefore the modification must be done again and again...

The right drives are configured by means of their device ID, see [Letters by Device ID](#) how to get it.

Sample for a USB drive:

```
[HideFromSafelyRemoveHardware]
DeviceID1=USB\VID_058F&PID_6369
```

Or by its "Friendly Name":

```
[HideFromSafelyRemoveHardware]
DeviceID1=Kingston Flash Reader
```

DeviceID and DeviceID1 to DeviceID9 can be configured.

You can also force a drive being shown under "Safely Remove Hardware" - USB DLM then set the removable flag in the registry:

```
[AddToSafelyRemoveHardware]
DeviceID1=IDE\DISKFUJITSU_MHV2040BH*00000025
```

But even you can get drives listed there, the safe removal facility usually fails to remove IDE drives, while my command-line tool [RemoveDrive](#) works.

Devices which are not related to drives can be handled too. But because USB DLM stores no data about other devices than drives, the complete device instance ID must be configured like so:

```
[HideFromSafelyRemoveHardware]
CompleteDeviceID=USB\VID_1758&PID_2004\050712014270000083
CompleteDeviceID=PCI\VEN_1011&DEV_0009&SUBSYS_21140A00&REV_20\3&267A616A&0&50
```

You find the device instance ID in the Windows Device Manager. Right-click the device -> Properties. Select the "Details" tab. Here the "Device Instance ID" is already selected under XP. Since Vista it is called "Device Instance Path" and must be selected from the properties list first. Then click on the ID and press "Ctrl+C" to copy the ID into the Windows Clipboard.

Device Description

In the Safe Removal Menu of XP and Vista USB drives are always called like "USB Mass Storage Device" (depending on the Windows language). USB drives are distinguished by their drive letter only. But drives mounted into NTFS folders or without a mountpoint have no further information, they are all called "USB Mass Storage Device" only.

But the name "USB Mass Storage Device" is read from the Registry value "DeviceDesc" which exists for each device. USB DLM can change this value:

```
[Settings]
SetDeviceDesc=1
```

Possible values:

```
SetDeviceDesc=0 -> no change
SetDeviceDesc=1 -> set for USB drives mounted to NTFS folders or unmounted
SetDeviceDesc=2 -> set for all USB drives
```

By default USB DLM writes the device's "FriendlyName" to the DeviceDesc value, something like "Corsair Flash Voyager". You can configure this using ~~USB DLM variables~~ `{linkID=}`.

Sample for FriendlyName and BusType for all USB drives:

```
[Settings]
SetDeviceDesc=2
DeviceDesc=%FriendlyName% USB Device
```

The result would be something like "Corsair Flash Voyager USB Device"

The value DeviceDesc is changed for the USB device. Since a USB device can have multiple volumes (either multislot cardreaders or multiple partitions) using volume specific variables makes no sense here, the last volume would "win".

The value DeviceDesc is also used by the Windows Device Manager!

Since Vista the DeviceDesc values have a new default content. For a USB drive it is

```
@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
```

All tries to replace certain parts only failed, only a full replacement succeed. I don't know which side effect

are caused by changing the value.

For other BusTypes than USB the name in the menu is not read from the DeviceDesc value. Probably here is no way to change the name.

Windows 7 shows the device name anyway and changing the DeviceDesc value has no effect on the display in the "Safely remove hardware" facility.

Passwords

Since V4.5 USBDLM can ask the user for a password on arrival of a drive or a media.

Hint: The OK button is the default button of the dialog, so you can just press Enter after entering the password.

Sample:

```
; restricted users are asked for a password
```

```
[Password]
```

```
UserGroup=Users
```

```
password=secret
```

```
; admins fall thru, so they are not asked for a password
```

If the password is noted in text form then you should remove read access for restricted users on the USBDLM.INI file...

Or note the password by its MD5 hash:

```
[Password]
```

```
UserGroup=Users
```

```
passwordMD5=5ebe2294ecd0e0f08eab7690d2a6ee69
```

[MD5-Text.zip](#)

The USBDLM service accepts one password per second, the number of tries is not limited.

When the user presses "Cancel" or closes the dialog then the drive is prepared for safe removal or it's media is ejected.

When the drive is removed while its password dialog is open, the dialog is closed.

Valid criteria for password sections:

- UserName
- UserGroup
- UserIsAdmin
- DeviceID
- BusType

In contrast to all other sections types the password sections are not limited to REMOVABLE and FIXED drives. So the user is asked for a password too when a floppy or CD/DVD drive is attached.

By means of the "USB DLM Remote Password Tool" the password can be sent over the network, so there is no need for the admin to tell someone the password...

[USB DLM-Remote.zip](#)

The password dialog at the client computer must stay open. As soon as the correct password is sent to the client the password dialog disappears there.

Sending the password relies on a former user authentication for instance by having a network share on the client computer.

Alternatively USB DLM can grant anonymous access to its password port:

[Settings]

RemotePasswordAnonymous=1

USB Write Protect

Windows since XP with SP2 can activate a write protection for USB drives. Unfortunately this is a global setting and not device specific.

Since V4.6.9.2 USBDLM can make this registry entry on arrival of a **USB device**.

Background information: When a USB drive is attached then the Windows plug'n play installs first the USB device, then it's disk device and finally the storage volume. Even the USB WriteProtect affects the volume only USBDLM sets the registry value already on arrival of the USB device to ensure that is ready when the volume is mounted. Therefore the only usable criteria is the device ID of the USB device.

The reg setting is the DWORD value "WriteProtect" under
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies.
1 activates the write protection for new USB volumes, 0 deactivates it.

USBDLM can set a 1 or a 0 for specific devices. USBDLM needs a default value to set for devices which no special setting is made for.

Sample:

```
; default: set a 0 for all devices which are not found  
; in sections [UsbWriteProtect] and [UsbNoWriteProtect]  
[Settings]  
UsbWriteProtect=0
```

```
; set a 1 for these devices  
[UsbWriteProtect]  
DeviceId1=USB\VID_13FE&PID_1F00
```

```
; set a 0 for these devices  
[UsbNoWriteProtect]  
DeviceId1=USB\VID_0951&PID_1624
```

You can also use an external text file with the IDs as shows for the [DeviceID](#) criteria.

The typical setting is to have UsbWriteProtect=1 and a section [UsbNoWriteProtect] with a list of drives which shall have no write protection.

The opposite way is UsbWriteProtect=0 and a section [UsbWriteProtect] with a list of drives which shall have a write protection.

The default value is also set when the USBDLM service is stopped.

The whole mechanism is not 100 percent reliable since it depends on the timing of the arrival event. If two USB drives with different WriteProtect settings are attached at the same time then the result is unpredictable. But usually it works fine.

Event Logging

USB DLM can write service start/stop, drive arrival, removal and AutoRun events to the Windows application log.

```
[Settings]
EventLog=1
```

It writes then items like these:

```
USB DLM service started
Drive attached: Name='FUJITSU MHK2120AT USB Device' Type=FIXED MountPoint='Z:\'
User=admin
Drive removed: Name='FUJITSU MHK2120AT USB Device' Type=FIXED MountPoint='Z:\'
User=admin
Open executed: MountPoint=Z:\ CmdLine=C:\Windows\System32\cmd.exe /c C:\Backup1.cmd Z:\
USB DLM service shut down
```

Since V4.3 this can be configured using [USB DLM variables](#).

Sample:

```
[Settings]
EventLogStringArrival=Drive attached: Name='%DevName%' Type=%DriveType%
MountPoint=%Root% User=%UserName%
EventLogStringRemoval=Drive removed : Name='%DevName%' Type=%DriveType%
MountPoint=%Root% User=%UserName%
EventLogStringOpen=Open executed: MountPoint=%Root% CmdLine=%CmdLine%
EventLogStringSvcStart=USB DLM service started
EventLogStringSvcStop=USB DLM service stopped
EventLogStringSvcShutdown=USB DLM service shut down
```

If an item is unwanted then set it like so:

```
EventLogStringOpen=-
```

Hint: While the Windows Event Viewer is open, the USB DLM.EXE cannot be deleted or replaced.

Since V5.2 USB DLM can write these lines to its log file with loglevel 0:

```
[Settings]
EventLog=2
```

If you want both then set EventLog=3

Prevent drives from going to sleep - Keep Alive

Some USB drives spin down after some idle time and then need a moment to answer on next access.

USB DLM can prevent the spin down by means of cyclic read accesses.

It is configured by means of [DiskKeepAlive] sections. Since it is related to disks and not to volumes, volume-specific criteria have no effect here.

Samples:

```
; read from all USB hard disks every 60 Seconds  
[DiskKeepAlive]  
DriveType=FIXED  
Interval=60000
```

```
; read from a certain USB hard disk every 60 Seconds  
[DiskKeepAlive]  
DeviceID=USB\VID_0781&PID_5580\AA01031913112811123  
Interval=60s
```

Default for the Interval is Null to make the Interval mandatory. Otherwise an empty section [DiskKeepAlive] would apply to all USB drives which would have been strange...

Auto-Dismount

Since V5.0 USBDLM can periodically try to dismount volumes (detaching the file system). This succeeds only if there is no open file on the volume. On next access Windows auto-mounts the file system again. This takes a moment, also because the file cache is discarded on dismount.

Knowing a volume being dismounted one could remove USB flash drives safely without ejecting or safely removing it.

Warning: Don't assume the removal of an SSD would be safe when the file system is dismounted! You never know if a SSD is performing optimization jobs internally. Therefore using "safe removal" is the only right choice here in hope that a shutdown signal makes it to the SSD.

USBDLM's auto-dismount works with local volumes only and for instance not with TrueCrypt volumes because they do not support the required notifications.

Problem:

The Windows Explorer's "Safely remove hardware" tray icon seems to freeze when the last drive in its menu is ejected by means of the Windows Explorer's drive context menu item "Eject" if the drive was already dismounted - by USBDLM or someone else.

To configure drives for auto-dismount create a section [AutoDismount] with criteria as for a DriveLetters section.

Since auto-dismount usually works with "removable" drives only (on fixed drives the Windows SearchIndexer makes it mounted again quite fast), DriveType=REMOVABLE would be the right criteria:

```
[AutoDismount]
DriveType=REMOVABLE
Interval=10s
```

Since Windows Vista on FIXED drives the Windows SearchIndexer is periodically accessing all volumes and causes them being mounted again. I've found no way to stop this other than disabling the "Windows Search" service (WSearch).

UsbDriveInfo shows the mount status.

An Interval of 0 disables the auto-dismount. This way you can exclude a drive:

```
; no auto-dismount for this drive
[AutoDismount]
DeviceID=USB\VID_13FE&PID_1F00
Interval=0
```

```
; for all other USB removable drives we want
```



```
[AutoDismount]
DriveType=REMOVABLE
Interval=10s
```

After arrival for some time no auto-dismount is performed, since usually the user wants to access the just arrived drive. This "grace period" is by default 30 seconds and can be configured globally:

```
[Settings]
AutoDismountGracePeriod=30s
```

Since V5.1 USB DLM can show tray icon with up to four drive letters of dismantled volumes:

```
[Settings]
AutoDismountIcon=2
```

0: Off
1: Show if there is at least one dismantled drive
2: Show always

Default is 2 if there is at least one [AutoDismount] section, 0 otherwise.

The background color of the tray icon can be configured with a Blue/Green/Red hex value like so:

```
[Settings]
AutoDismountIconBackgroundColor=0x80FF80
```

For now only drive letters can be shown in the tray icon and only up to four in alphabetic order.

On a mouse-move over the tray icon all volumes configured for auto-dismount are tried to dismantl immediately, therefore an always visible tray-icon is useful.

If you want no auto-dismount but manual only then just configure -1 as interval, this resolves to an interval of 232 milliseconds (49.7 days).

BadUSB Device Blocking

Since many years it is a known threat that a bad USB device can act as a USB keyboard and therefore send keystrokes to Windows for doing bad things as downloading and executing malware from the internet.

In 2014 the German company [Security Research Labs](#) showed at the Black Hat conference that it is possible to reprogram widely spread USB controller chips also used in flash drives to turn them into evil "[BadUSB](#)" devices. Meanwhile Adam Caudill and Brandon Wilson have [published](#) all required tools for creating a BadUSB device. This is a new quality of the problem.

The main attack scenario is a USB pen which acts as a keyboard device so it can issue commands on behalf of the logged-in user.

Under Windows XP this can be completely blocked by moving or changing the file extension of the C:\Windows\inf\keyboard.inf file. As long this is not present there will be no new keyboard device installed.

Since Vista new devices of certain device classes can be blocked, just search for [DenyDeviceClasses](#). Here is a REG file which blocks all new keyboard devices under Vista, Win7 etc: [DenyNewKeyboards.reg](#)

A bit more handy is what USB DLM does: It prepares a new USB keyboard device for safe removal immediately after arrival and asks the user if it shall be activated again. There is a short moment while keystrokes are accepted but this is by far not long enough to open a command prompt and write and start an evil script.

There is no check if there is any other input device which of course the user needs to answer the question. If required the user can bring a USB mouse...

[Settings]

BadUsbWatchKbd=1

- BadUsbWatchKbd=0 -> USB keyboard devices are installed normally (default)
- BadUsbWatchKbd=1 -> The user is asked on arrival of a USB keyboard device
- BadUsbWatchKbd=2 -> On arrival any USB keyboard device is silently prepared for safe removal

Of course the remaining attack scenario is a real USB keyboard which can perform evil key sequences. But this is true since keyboards exist, USB or not.

There is also talk about USB network adaptors which could manipulate DNS requests and/or the IP-Routing. USB DLM can block them too, the setting works equally to BadUsbWatchKbd:

[Settings]

BadUsbWatchNet=1

Title and text of the message box can be changed. Sample:

[Settings]

UsbKbdMsgTitle=USBDLM

UsbKbdMsgText=USB keyboard device attached. Activate?

UsbNetMsgTitle=USBDLM

UsbNetMsgText=USB **network** device attached. Activate?

White and blacklisting of device ids and USB ports

To not being asked again and again for known non-evil devices, e.g. a WiFi adapter, you can whitelist them by device id. Sample

[BadUsbWhiteList]

DeviceID1=USB\VID_057C&PID_5601

The device ID must be the one of the device with the device class keyboard or net respectively.

[UsbTreeView](#) shows it.

Sample for what UsbTreeView shows for a USB composite device with an HID keyboard:

```
Child Device 1      : USB-HID (Human Interface Device)
Device ID          : USB\VID_05FE&PID_2001&MI_00\8&1D9FC75B&0&0000
Class              : HIDClass
Child Device 1      : HID Keyboard
Device ID          : HID\VID_05FE&PID_2001&MI_00\9&2594D27D&0&0000
Class              : Keyboard
```

Use the bold part only because the last part changes on each port if the device has no USB serial number.

Furthermore it might be handy for instance to say that real USB keyboards are attached at the back side ports of an PC, so these are whitelisted. Sample:

[BadUsb**White**List]

UsbPort1=1-1

UsbPort2=1-2

UsbPort3=1-3

UsbPort4=1-4

UsbPort5=1-5

UsbPort6=1-6

Or you say that keyboard or network devices are never ever attached to the front ports. Sample:

[BadUsb**Black**List]

UsbPort1=1-7

UsbPort2=1-8

For blacklisted devices or ports the user is not asked, the devices are silently prepared for safe removal. The blacklist is checked first, so it has a higher priority than the whitelist.

To determine the port names you have to attach a USB drive to each port and check the port name by means

of the UsbDriveInfo tool. For non-USB devices you need [UsbTreeView](#).

The first digit of the port name is the index of the root hub which can change when a USB controller is added or removed. Use the long notation then, see [Letters by USB Port](#). Furthermore most USB2 controllers use separate logical root hubs for USB1 devices, so the index is different here too.

Deficits:

USBDM handles newly attached devices only. If a BadUSB device is attached before logon then it will work normally but will play its bad keystrokes on the logon screen where they will not work.

The problem with present devices is that USBDM cannot know which one is the regular keyboard.

Disabling it would lock out the user..

White and Blacklists work too if no user is logged on and the USBDM service already running on arrival of the device.

Plans:

- MessageBox at a random position if the USB device comes with a mouse too and thus could click the Yes button on the MessageBox

Links:

[BadUSB article at arstechnica/](#)

[Presentation at Black Hat 2005 - Barrall-Dewey](#)

After reading the Presentation from Black Hat 2005 you know that the problem isn't new at all and that USBDM can block only the most simple USB attacks.

Hopefully Microsoft has hardened all their drivers involved with USB devices against these attacks...

XP System Restore

The XP System Restore wants to watch external harddrives too even this is completely useless and only great for screwing up system restore points by not having the same disks attached when required. Manually disabling System Restore for a drive is lost when a restore point is created while the drive isn't attached. In practice it must be disabled again and again for USB hard drives.

Since Vista only the system drive is watched.

USB DLM can try to disable the system restore for USB and Firewire drives but this is experimental. In most cases it works but sometimes it does not.

[Settings]

DisableSystemRestoreForNewDrives=1

Furthermore USB DLM can try to delete the beloved "System Volume Information" folder:

[Settings]

DeleteSviFolder=1

Caution: If the drive contains data of a system restore point of a foreign XP computer then these data are lost of course and this system restore point is screwed up. But this is the case too if the drive is just not there...

If the delete attempt fails then USB DLM can stop the System Restore Service for a moment and try again:

[Settings]

DeleteSviFolder=2

To prevent at least the system restore scans around the disk and fills the "System Volume Information" folder, USB DLM can add a value to the registry which prevents files and folders being backed up by NT-Backup and the system restore.

<http://www.tweakxp.com/article37472.aspx>

[Settings]

AddNewDrivesToFilesNotToBackup=1

Vista / Windows 7 "Scan and Fix"

When a FAT formatted "removable" USB drive is attached then under Vista and Win7 the "Scan and Fix" dialog is often shown.

After executing this "Scan and Fix" (never seen it really finds an error) it shall not be shown again, but in real life it happens that it comes again and again...

Why?

A look at the FAT partition's boot sector reveals that the "Scan and Fix" is shown when there is a certain bit set. When this bit is set back to 0 then "Scan and Fix" is not shown anymore.

USB DLM can do exactly this:

[Settings]

```
PreventVistaScanAndFix=1
```

What exactly does USB DLM?

This aforesaid bit is found in a value with the very meaningful name "BS_Reserved1", see Microsoft's [FAT32 File System Specification](#) (fatgen103.doc).

Usually its value is 0, when "Scan and Fix" was triggered you will find a 1 there.

When a new USB drive is attached then USB DLM looks for a FAT partition and resets the value to 0 in its bootsector. If this succeeds before Windows reads the value then "Scan and Fix" is not shown. Usually it works well, but sometimes Windows is too fast...

Of course this suppresses the symptom only. It is far better to prevent having this "dirty bit" set.

As far as I have discovered, Windows sets the bit whenever a file size is changed and when a file is created or deleted.

Here is my dirty bit watch tool: [WatchFatDirtyBit.zip](#)

Under Windows 7 it is set back to 0 about 1.5 Seconds after a file size change but very long 30 Seconds after a file was deleted or created! Having a write cache active or not seems to have no effect on the dirty bit handling under Windows 7. But this there are very first results, more research is to come...

So, if you delete a file on a FAT formatted USB drive and remove it within 30 Seconds then you have set dirty bit.

Just always use "Eject" or the "Safely remove hardware" facility, this flushes all data and sets the bit back to null.

TrueCrypt / VeraCrypt

TrueCrypt is an open source software for creating encrypted drives. The drive's data is hold by a "container". This container can be a file, a whole disk partition or a whole disk.

<https://www.grc.com/misc/truecrypt/truecrypt.htm>

In May 2014 the TrueCrypt developers lost interest in developing the software and declared it as potential unsafe because "it may contain unfixed security issues". An audit of the TrueCrypt source code didn't reveal any real issues, so V7.1 can be assumed to be safe.

Based upon the TrueCrypt source code there is the VeraCrypt project, functionality, usage and command-line parameters are quite the same as with TrueCrypt. Only the mounting of a volume is very slow to make it safe against brute force attacks. With the latest release the hash algorithm can be selected on mounting, this makes the mounting fast enough.

<https://veracrypt.codeplex.com>

If TrueCrypt is mentioned in the following then it means VeraCrypt too.

A TrueCrypt container contains random data which is used for holding the encrypted data. From the outer point of view the container has no file system, no volume label. Nothing else than random data.

Only the TrueCrypt application with its driver is able to mount the container and make its contents available decrypted under an additional drive letter.

Whole drive as Container

The first sector of a drive is expected to contain a partition table or a boot sector. Therefore it is not a good idea to use a complete drive as a TrueCrypt container. When opening the Disk Management Windows might want to "initialize" the drive which is not so good for the TrueCrypt container. On the other hand the Windows Explorer asks to format a TrueCrypt container volume, which is not less dangerous.

Since V4.8 USBDM has the new events OnDiskArrival, OnDiskRemovalRequest and OnDiskRemoval, so whole disks used as TC container can be handled too.

On disk arrival the disk number is known (it's the "device number") and Partition0 stands for the whole disk under Windows.

```
;mount it on J:  
[OnDiskArrival]  
DeviceType=TrueCrypt  
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v \Device\Harddisk%DeviceNumber\Partition0  
/letter J
```

On removal the DeviceType can be ReadSharingViolation or TrueCrypt, depending on if the volume was

already mounted when the USBDLM service started.

```
;dismount J: on safe removal of the disk  
[OnDiskRemovalRequest]  
DeviceTypes=TrueCrypt,ReadSharingViolation  
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dJ
```

Instead of the TrueCrypt tool you can use my command-line tool EjectTcv to eject the TrueCrypt volume:
http://www.uwe-sieber.de/drivetools_e.html#ejecttcv

It can do it by a given container name and in a loop until success. Sample:

```
;dismount the TrueCrypt volume on safe removal of the disk, EjectTcv can do by container name  
[OnDiskRemovalRequest]  
DeviceTypes=TrueCrypt,ReadSharingViolation  
open="C:\Tools\EjectTcv.exe" \Device\Harddisk%DeviceNumber%\Partition0 -L
```

Partition as Container

Best with an unknown partition type

If a whole partition is used as TrueCrypt container, then Windows will assign a drive letter anyway. This is useless and dangerous. The drive will be shown as having a file system "RAW" and Windows may give the suggestion to format it.

You can prevent this by patching the partition type to one Windows doesn't know, e.g. 0x7C. Windows still enumerates the partition but it does not create a volume device for it, so it is safe from being formatted and no drive letter is assigned. Without a volume device USBDLM gets the disk arrival notification only, so you have to use OnDiskArrival as shown above but with the right partition number instead of 0 for the whole disk.

Changing a partition type can be done by means of the [Active@ Partition Manager 3.0](#): Right-click on the drive, Partition Table..., here set the partition's value "File system [hex]" on 7C.

With a standard partition type

With a standard partition type Windows creates a volume device and assigns a drive letter to the TrueCrypt container. Selected in the Windows Explorer it suggests to format it... Therefore it is a good idea to remove the drive letter of a partition used as TrueCrypt container. This can be done in the Windows Disk Management, but as all assignments made here stick exactly for the drive they are made for only.

USBDLM can remove the drive letter of any partition used as TrueCrypt container. In fact USBDLM checks the partition's first sector for random data. If random data is found, USBDLM gives the volume the [DeviceType](#) "TrueCrypt". If the container is already mounted by TrueCrypt then USBDLM cannot check it for random data since TrueCrypt holds the container open with exclusive access. The attempt to read is rejected with the error code "ERROR_SHARING_VIOLATION". That's why USBDLM gives the DeviceType "ReadSharingViolation" then. This usually happens when the USBDLM service is manually restarted.

Typical handling of a TrueCrypt container partition: Remove the container's drive letter and call TrueCrypt

to mount it.

If the USB DLM service is restarted while the TrueCrypt container is mounted, then we want the letter for the container partition removed but the volume shall not be tried to mount again by TrueCrypt, that's what this sample does:

```
;remove the volume's drive letter, it's useless
```

```
[DriveLetters]
```

```
DeviceType1=TrueCrypt
```

```
DeviceType2=ReadSharingViolation
```

```
Letter=-
```

```
;mount it on J:
```

```
[OnArrival]
```

```
DeviceType=TrueCrypt
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v %PartitionName% /letter J
```

```
;an Explorer window with the mounted TrueCrypt volume
```

```
[OnArrival]
```

```
DeviceType=TrueCryptVolume
```

```
open="%windir%\explorer.exe" %root%
```

%PartitionName% is a USB DLM variable {linkID=} which USB DLM replaces with something like \Device\Harddisk2\Partition1 as expected by TrueCrypt.

File as Container

When a drive arrives then USB DLM can check for the presence of a file and let TrueCrypt mount it:

```
[OnArrival]
```

```
FileExists=%drive%\secret.tc
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\secret.tc" /letter J
```

Maybe check if the target letter is available by means of the DirNotExists criteria, but then it is silently not mounted instead TrueCrypt showing a drive letter not available error message:

```
[OnArrival]
```

```
FileExists=%drive%\secret.tc
```

```
DirNotExists=J:\
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\secret.tc" /letter J
```

[Since V7.0](#) TrueCrypt can automatically mount containers itself.

The TrueCrypt Volume

When TrueCrypt mounts a container, then it makes its contents available under an additional drive letter. This drive isn't a full blown "storage volume", it is implemented partially only.

USBDM sees the arrival of the drive letter and can trigger an OnArrival for it. The TrueCrypt drive's DeviceType is "TrueCryptVolume" ("VeraCryptVolume" resp). That's all. USBDM knows its drive letter and makes it available in the USBDM variables as %drive%. But it cannot deal with the drive's drive letter. In fact it could but TrueCrypt would lose track.

By means of the Windows tool MOUNTVOL the TrueCrypt volume can be mounted into an existing folder on an NTFS drive:

```
[OnArrival]
```

```
DeviceType=TrueCryptVolume
```

```
open1="%windir%\System32\mountvol.exe" "C:\TrueCrypt" %VolumeName%
```

```
system1=1
```

```
open2="%windir%\explorer.exe" %root%
```

USBDM does not know anything about the relation between TrueCrypt container and the mounted TrueCrypt drive. It just handles subsequent events.

Safe Removal

On a request for "Safe Removal" of the host drive, USBDM can let unmount a TrueCrypt volume.

Unfortunately TrueCrypt can unmount by drive letter only.

Since USBDM V5.0 there is the criteria IsMountedTrueCryptContainer, which can check if a container (file, partition, disk) is mounted by TrueCrypt:

; check if secret.tc on the drive to remove is mounted by TrueCrypt

```
[OnRemovalRequest]
```

```
IsMountedTrueCryptContainer=%drive%\secret.tc
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dismount J
```

; check if the volume to on the drive to remove is mounted by TrueCrypt

```
[OnRemovalRequest]
```

```
IsMountedTrueCryptContainer=%PartitionName%
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dismount J
```

If the partition type is patched the an unknown one then there is no OnRemovalRequest but a OnDiskRemovalRequest. We have to know and build the partition name ourself because there is no volume:

; check if partition 1 on the drive to remove is mounted by TrueCrypt

```
[OnDiskRemovalRequest]
```

```
IsMountedTrueCryptContainer=\Device\Harddisk%DeviceNumber%\Partition1
```

```
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dismount J
```

The parameter /q or /quit is important because USBDM answers the removal request not before the started TrueCrypt.exe ends. Without /quit TrueCrypt shows a window and USBDM waits and waits... finally Windows will say "The service USBDM prevents the removal".

For releasing a TrueCrypt volume I have made a command-line tool which accepts the TrueCrypt volume's

container name as drive spec. Furthermore it can try again until success.

```
; EjectTcv can unmount by container name
```

```
[OnRemovalRequest]
```

```
IsMountedTrueCryptContainer=%drive%\secret.tc
```

```
open="C:\Tools\EjectTcv.exe" %drive%\secret.tc -L -H
```

http://www.uwe-sieber.de/drivetools_e.html#ejecttcv

BitLocker To Go

BitLocker is a full storage volume encryption introduced with Windows Vista. But BitLocker supports local disks only.

With Windows 7 came "BitLocker To Go" which allows to encrypt hotplug drives. It comes with the Ultimate and Enterprise Edition only.

BitLocker works with a "Filter Driver" which blocks or grants access to the volume.

A hotplug volume encrypted with BitLocker comes in two different shapes. It depends on the drive's file system which one we get.

BitLocker encrypted FAT/FAT32/exFAT drives still have an unencrypted FAT file system. It contains a lot of files, most of them have the "hidden" attribute, so most users will not see them. Two files are not hidden: The autorun.inf and the BitLockerToGo.exe.

Thru the autorun.inf the BitLockerToGo.exe shall be started, otherwise you can start the BitLockerToGo.exe manually. The BitLockerToGo.exe opens an Explorer like window which gives read access to the BitLocker volume only.

BitLocker encrypted NTFS drives don't have this feature. On Windows without BitLocker To Go support Windows suggests to format the drives.

What USB DLM can do...

1) On Windows with support for "BitLocker To Go"

Encrypted volumes are detected as such, they get the DeviceType BitLocker, it is removed as soon as the drive is unlocked.

Remounting is done immediately, volume specific criteria do not work because the volume cannot be read yet. The BalloonTip is shown immediately too.

AutoRun events are triggered when the volume is unlocked.

2) On Windows without support for "BitLocker To Go"

"BitLocker To Go" encrypted NTFS volumes get the DeviceType BitLocker.

We can remove the drive letter to avoid the user follow Windows' suggestion to format the drive

[Driveletters]

DeviceType=BitLocker

Letter=-

Furthermore we can show a manually made BalloonTip which informs the user about the problem:

```
[OnArrival]
```

```
DeviceType=BitLocker
```

```
open="%usbdlmpath%\usbdlm.exe" -balloon -time=20000 -title="%FriendlyName%" -text1="Cannot read  
BitLocker encrypted volume" -icon=
```

"BitLocker To Go" encrypted FAT volumes have regular FAT file system, therefore they do not get the DeviceType BitLocker.

Here we can look at the presence of the BitLockerToGo.exe and execute it:

```
[OnArrival]
```

```
FileExists=%drive%\BitLockerToGo.exe
```

```
open=%drive%\BitLockerToGo.exe
```

Problems

Wrong drive letter for a short moment

When a drive is attached then it usually comes with a drive letter which (if required) USBDLM changes at the earliest possible point. If an application reads the drive letter at the same point then it gets confused when USBDLM changes the letter a millisecond later. This is a very rare problem since most applications use a later notification.

To fix this problem, after removal of a drive, USBDLM can create a registry entry which makes a drive come up without a drive letter assigned next time, so, USBDLM can assign one then and no one gets confused. But this is usually not required.

[Settings]

```
DeleteLettersOnRemoval=1
```

Since V4.8.6 USBDLM tries to mess with the Windows mount manager by writing a no-letter entry on disk arrival, a moment before the volume arrives. This works good up to Windows 7. Windows 8 is too fast. At least USBDLM can delete the wrong letter a bit earlier than before.

USB DLM service starts to late

Since XP Windows does not delay the user logon until all services are started - the illusion of a fast system start has priority...

So it can happen that the user get logged on before the USBDLM service is started. In result network drives might fail to create if their drive letter is used and USBDLM starts too late to fix this.

Since V4.5.1 USBDLM can delay the execution of AutoRuns and the loading of the Windows Explorer until it is started:

[Settings]

```
DelayDesktop=1
```

This works very simple: On logoff of the user USBDLM creates a RunOnce entry in the Windows Registry which starts the USBDLM.EXE with a special parameter. This USBDLM instance then waits for the USBDLM service reaches status "STARTED" and ends. Afterwards Windows continues with the normal startup process by starting the Windows Explorer and executing all the auto startup stuff. The startup might feel slower then but in fact it is just a different loading order.

Windows Explorer fails when deleting a folder under XP

This is one of many issues when a drive is mounted into an NTFS folder, it's not a USBDLM issue. Please read "[Mounting to NTFS folders](#)".

Drive letters accidental reassigned

Drive letters removed by NoMediaNoLetter or when drives are mounted into NTFS folders are reassigned when the "U3 launchpad" (the U3launch.exe on the fake CDROM drive or installed in C:\Documents and Settings\All Users\Application Data\U3) is started. It's an eternal bug in the U3 software. USBDLM since V4.3 can try to fight this when configured:

[Settings]

FightU3Bug=1

But it's not 100 percent reliable.

A more strong option is

[Settings]

ForceNoMediaNoLetter=1

Another software which is known to assign drive letters is "Secure Storage Device SDK" (SSDService.exe) from MXI.

Cannot delete or replace the USBDLM.EXE

While the Windows Event Viewer is open, the USBDLM.EXE cannot be deleted or replaced. This is by design.

Devices which failed to start under Vista, Windows 7

For USB removable drives Vista and Windows 7 always install a "WPD File System". WPD is "Windows Portable Device", a programming interface for devices like MP3-players, mobile phones etc. Since Windows 7 the entries are found under "Portable Devices" in the Windows Device Manager.

If a USB removable drive has no drive letter (no matter if it was USBDLM which removed it or the Windows Disk Management), the start of the associated WPD device fails with "Code 10". This is ugly but no problem - the drive works anyway just as a drive.

If you just want to get rid of the WPD stuff, you can deactivate (or set to manual start) the "Portable Device Enumerator Service" (WPDBusEnum). A nice side effect is that the Windows Media Player stops then synchronizing with USB mass storage devices.

Since Windows 8 the problem is fixed.

Vista / Win7 / Win8 Virtual Store

The "User Account Control" (UAC), introduced with Windows Vista, denies write access to system critical data even to the admin which includes all files under "C:\Program Files". Therefore programs which want to write to an INI file located there would have a problem. Microsoft's solution is to redirect INI files to the "Virtual Store" which is a folder under the current user's documents folder, e.g.

C:\Users/(UserName)/AppData/Local/Virtual Store/Program Files/(Program).

Windows creates a copy of the original INI file on first write access. As a result, there are multiple INI files, the original and one for each user.

If an INI file exists in the Virtual Store then this one is used when accessed by a logged on user.

The USBDLM service is not affected by the UAC, therefore it read from its own folder. But the user who

edits the INI file is redirected to the Virtual Store...

Since V4.3 USBDLM shows a warning balloon tip if a USBDLM.INI is found in the current user's Virtual Store. A click on it opens the Virtual Store's folder in the Windows Explorer.

Solutions:

- never forget to edit the USBDLM.INI as a real admin ("elevated", "As Administrator"), since V4.7.3 there is the `_edit-ini.cmd` to do so
- install USBDLM into a different folder, e.g. C:\USBDLM or C:\Tools\USBDLM. The Virtual Store deals with "C:\Program Files" only. In this case remove write access rights to this folder for non admins!

Windows 2000

When a drive is attached for the first time, then it appears in the system, then it disappears and appears again. Since XP USBDLM can deal with it, under Windows 2000 some workarounds are required. If the workarounds fail then the balloontip and AutoRun might be skipped or executed twice.

TrueCrypt drives

Drives used as TrueCrypt containers can be detected as such only as long as they are not mounted by TrueCrypt. This is because TrueCrypt takes exclusive access to the drive so USBDLM cannot read test data. This is a problem for UsbDriveInfo only and when the USBDLM service is restarted while a TrueCrypt drive is already mounted. It is detected then with a DeviceType ReadSharingViolation instead of TrueCrypt. Read more in section [TrueCrypt](#).

Error 193

If the start of the USBDLM service fails with error 193 (which is `ERROR_BAD_EXE_FORMAT`), then something is wrong with the USBDLM.EXE. Either is in a folder which cannot be read by the "SYSTEM" account which is required for a service being started by the Service Control Manager. Fix it by giving "SYSTEM" at least read+execute access. Or you try to use the x64 USBDLM.exe under a 32 bit Windows.

Technical Information

Here some background information, just what came into my mind...

Window Service

USBDM is a Windows Service. As such it is registered in the registry and started and stopped by the "Service Control Manager".

Services usually are running in the context "LocalSystem" and have highest privileges here. USBDM is a Win32 service, not a kernel service. A kernel service could have the start type "LocalSystem" which would ensure that it is loaded before the user logs on. For a Win32 service "Automatic" is earliest start type. With "Automatic" the user might be logged on before the services starts, see also "USBDM service starts to late" under [Problems](#).

BalloonTips

Under Windows 2000 services have no problem showing a window on the user's desktop. All they need it the "interactive" flag. Under XP this works for the first logged on user only. Further users are running in a different "Session" with its own desktop.

Under Vista all users are running in their own session, isolated from the services. Services with the "interactive" flag are punished with a warning in the system log...

If the user is running in the same session then USBDM shows the balloons directly. Otherwise it starts itself in the user's context and shows the balloon.

You can remove the "interactive" flag, USBDM will show the balloons then indirectly.

The USBDM.MSI has the "interactivte" Flag but USBDM removes it when it is started first under Vista and higher.

When USBDM starts itself for showing a balloontip, it gives the data for the balloon tip at the command-line. This can be used for custom balloon tips, e.g. on removal of a drive:

[OnRemoval]

```
open="%usbdlmpath%\usbdlm.exe" -balloon -time=1500 -title="%FriendlyName% %BusType% Device" -text1="removed: %root%" -icon=
```

But such balloon tips are completely independent from the regular USBDM balloons, so multiple balloons can be shown at the same time, BalloonClicks do not work etc...

The balloon tips are handmade. The main reason is that under W2K and XP the maximum text length is too limited to show large balloons, e.g. with multiple NTFS mountpoints. Furthermore with the handmade

balloons we don't need a tray-icon and the taskbar does not pop up when the balloon is shown.

Available icon numbers are:

110 one drive

112 two drives

113 one CDROM

114 one network drive

120 one drive removed

121 one drive safely removed

122 one drive unsafely removed

123 one CDROM drive removed

124 one network drive removed

130 an information sign

131 an exclamation mark

Message Boxes

The USBDLM_usr.exe can show message boxes like so:

[OnUnsafeRemoval]

```
open="%USBDLMPATH%\USB DLM_usr.exe" -msgbox -type:0 -icon:0x30 -title:Warning -
text:%BusType% device %FriendlyName% was unsafely removed. This is not safe!
```

Type, icon and return value (aka errorlevel) are as with the Windows API call [MessageBox](#).

Type refers to the shown buttons as MB_OK (0..6), icon to the shown icon as MB_ICONERROR (0x10, 0x20, 0x30 or 0x40).

TrueCrypt, Network and Subst drives

TrueCrypt volumes are not fully implemented. For instance there is no notification when the volume is mounted. This is true for Network and Subst drives too.

But even under Windows 10 there is still the old broadcast notification sent to all windows when a drive letter arrives or a CD is inserted into a CD drive. This comes too when a TrueCrypt volume is mounted, a Network or Subst drives is created.

To receive this USBDLM has an invisible window. But this does not work when the user is running in a different session which is always true under Vista and Windows 7+. Under XP it is true when more than one user logs on at the same time.

USB DLM needs a helper application then which receives the message and passes it "down" to the service.

That's what the USB DLM_usr.exe is for. It is loaded when the user is running in a different session and one

of this points is true:

- ExternalMountEvents=1
- an [OnArrival] with DeviceType=TrueCryptVolume
- an [OnArrival] with DeviceType=SubstDrive
- an [OnArrival] with BusType=remote
- [BalloonTips] ShowTrueCrypt=1
- [BalloonTips] ShowVirtual=1
- USBDLM Win32 under Vista x64 with NoMediaNoLetter=1 (workaround for [Windows-Bug in the WoW64 layer](#))

Under Vista and Windows 7+ we deal with the User Account Control, the UAC. Here the user context is split into a restricted and an unrestricted one (the elevated one). To receive notifications for both we need the USBDLM_usr running twice if:

- an [OnArrival] with DeviceType=SubstDrive and elevated=1
- an [OnArrival] with BusType=remote and elevated=1
- [BalloonTips] ShowTrueCrypt=2 or ShowTrueVirtual=2

Creating an MSI

Having USBDLM as an MSI with included USBDLM.INI file is useful for distributing USBDLM in a network by means of a group policy, admins know better than I do...

The USBDLM.MSI provided up to V4.5.9 is created by means of OnDemand WinInstall LE.

Bug: For all versions I used identical product and update GUIDs. The result is that you can use any version to uninstall another, but updates do not work. So, for an update it is necessary to uninstall the previous version and then install the new one.

Beginning with V4.6 the Microsoft WIX toolset is used for creating the MSI. Here correct GUIDs are used, so updates work but for uninstalling the exact version used for installing is required.

Updating very old version of USBDLM please install V4.4 first, then the latest version. The reason is that for stopping the USBDLM service USBDLM's command-line parameter `-silentstop` is used. This was not available until V4.4.

You need the [Microsoft WIX](#) toolset. Download the WIX3x.MSI, right click -> Install.

If there is no Visual Studio installed it may come first with a warning message. Just click OK, its about the Visual Studio integration only.

WiX installs to the default programs folder, e.g. to "C:\Program Files\WiX Toolset v3.8", the folder where the environment variable `%ProgramFiles%` points to.

WiX requires the [.NET 3.5 Framework](#).

Under Win7 the .NET 3.5 framework can be activated in the Control Panel -> Programs and Features -> Turn Windows feature on or off.

Or by command-line:

```
DISM /Online /Enable-Feature /FeatureName:NetFx3 /ALL /NoRestart
```

Here is the WIX script for USBDLM: [USB DLM.WXS.ZIP](#)

In this script there is a line for adding the "USB DLM_sample.ini", change this into "USB DLM.INI" if you need one.

Put the files from the USB DLM.ZIP archive into the USB DLM folder, the USB DLM.WXS, your customized USB DLM.INI and a `make_msi.cmd` like this:

```
@echo off
```

```
set WiX_Path=%ProgramFiles%\WiX Toolset v3.8\bin
```

```
if not exist "%WiX_Path%" echo !!!!!!!!!!!!! WiX not found !!!!!!!!!!!!!!! & pause & goto ExitPoint
```

```
rem -- ensure the folder of the CMD file is the current folder --
```

```
cd /D %~dp0
```

```
rem -- execute WiX candle and light, pause & exit if error --  
"%WiX_Path%\candle.exe" -dProcessorArchitecture=x86 -arch x86 -out USBDLM.wixobj USBDLM.wxs  
if errorlevel 1 pause & goto ExitPoint
```

```
"%WiX_Path%\light.exe" -out USBDLM.msi USBDLM.wixobj  
if errorlevel 1 pause & goto ExitPoint
```

```
rem -- cleanup --  
del USBDLM.wixobj  
del USBDLM.wixpdb
```

```
:ExitPoint
```

For the x64 version change x86 to x64:

```
"%WiX_Path%\candle.exe" -dProcessorArchitecture=x64 -arch x64 -out USBDLM.wixobj USBDLM.wxs
```

The wixpdb file created where the MSI is created, so, if you change -out to another folder then put this path to the "del USBDLM.wixpdb" line too.