Index

USB Drive Letter Manager V4.4.1

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

USBDLM is a Windows service that gives control over Window's drive letter assignment for USB drives. 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.
In can deal with non USB drives too when a BusType or DeviceID configured.

USBDLM works with Windows 2000, XP, Server 2003, Vista, Windows 7 and Server 2008. The x64 version works on x64 editions of Windows only.

It is tested under

- Windows 2000 SP4 + Rollup 1

- Windows XP SP2
- Windows XP SP3
- Windows XP x64 SP1
- Windows 2003 Server SP1
- Windows Vista Bussiness N SP2
- Windows Vista Bussiness N x64 SP1
- Windows 7 Ultimate x64
- Windows 2008 Server

Installation

USBDLM does not come with a SETUP.EXE, it's all manual work. See Section Installation.

Conditions

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

USBDLM 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/usbdlm_e.html

Licence agreement

Why?

When a removable drive (USB flash drive, flash card reader, portable hard drive) is attached for the first time, Windows mounts it to the first available drive letter. If there is a network share on this letter, then Windows XP-SP2 will use it anyway for the new USB drive because since XP network shares are specific to the current user and not visible in the context of the system where the letter is assigned. The USB drive then appears to be invisible.
You can change the letter assignments in the Windows Disk Management Console with a lot of mouse clicks but you have to do it again for every new device.
And, for USB devices that have no serial number you have to do it too when you attach it to a different USB port.

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 dependend 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
- let show an balloontip on drive arrival which show the assigned drive letter
- 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.

Installation

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

Under Vista and Windows 7 I suggest to install USBDLM in a different folder than `C:\Program Files,` because under this folder changes of the USBDLM.INI made by a non elevated user are redirected into the "Vista Virtual Store" where USBDLM will not read it.
In a folder like `C:\Tools\USBDLM` there is no problem.

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

Then enter your prefered configuration into the USBDLM.INI (see below). There is a sample file included called USBDLM_sample.INI. Either create a new USBDLM.INI or rename the sample file.
On an NTFS formatted drive editing the USBDLM.INI requires admin privileges. Under Vista you should start the Editor elevated, otherwise the USBDLM.INI might be redirected into the Vista virtual store where it has no effect.

If you are logged on with administrator privileges you can simply install the USBDLM service.

```
_install.cmd
```

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

Once installed USBDLM 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 USBDLM.EXE is loaded from the folder where the _install.cmd had been started from!

Uninstallation:
You can uninstall the service using parameter -uninstall:

```
_uninstall.cmd
```

This stops the service and unregisters it.

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

```
USBDLM -silentinstall
USBDLM -silentuninstall
```

It returns Errorlevel 0 on success, 1 on failure.

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

```
net stop USBDLM
net start USBDLM
```

or by means of the _stop and _start command script which are using the USBDLM.EXE.

You can change the drive letter settings in the USBDLM.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 only like
`USBDLM -install`
Therefore they must be started from the USBDLM folder, otherwise the USBDLM.EXE isn't found. To make the CMD file work when started from another folder modify them this way:
`"%~dp0USBDLM" -install`

`%~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\USBDLM\_install.cmd`.
`~dp` is a "modifier" which delivers drive and path like `C:\Program Files\USBDLM`. 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 are defined in a text file called `USBDLM.INI` located in the same place as the USBDLM.EXE.
Modern applications often stores its settings in the Windows registry but I don't like that. INI files are the 'classic' approach.

Windows Vista, Windows 7:
Since Vista the "Virtual Store" might redirect the INI file to
`C:\Users\(UserName)\AppData\Local\VirtualStore\Program Files\USBDLM`
But the USBDLM service always reads from its own folder. So, under Vista always edit the INI file as "real" Administrator.
Or install USBDLM to a different folder, for instance to C:\Tools\USBDLM because the Vista Virtual Store deals with the "C:\Program Files" folder only.

The "correct" data folder for Win32 services as USBDLM differs depending on the Windows version and language. On an English XP it is "C:\Documents and Settings\LocalService\Application Data\USBDLM". Under Vista it is "C:\Windows\System32\config\systemprofile\AppData\Roaming\USBDLM". But under Vista x64 "C:\Windows\System32" is redirected to "C:\Windows\SysWOW64" for Win32 programs.
Since V4.3 searches the USBDLM.INI first in this folder.

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.
Each section name is allowed to exist once only! Same thing for the values within a section.
Comment lines begin with a semicolon. Never try to deactivate a section this way:

```
;[DriveLetters20]
```

This extends the previous section to the begin of the next section which leads to unexpected results. This is the right way:

```
[xxxx DriveLetters20]
```

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

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/usbdlm_sample.reg

Under x64 Windows the Win32 edition of USBDLM is redirected to
`HKLM/Software/Wow6432Node/Uwe Sieber/USBDLM`

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 letters.
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:

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

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

If you want to configure a single letter only, then you can omit the number:

```
Letter=X
```

This is not made to be used mixed with numbered lines but if you do, then the line without a number is read first.

Since V4.1 you can use a **short notation** in lines "Letter**s**="**.** 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 handle equal, so you can even write `Letter1=AWXYZ.`

By default USBDLM 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:

```
[DriveLetters10]
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.

Prioriy

First "Letters" is read, then "Letter" and finally Letter1 to Letter9.

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 an USB disk at X: and hide partition 2:

```
[DriveLetters]
Letter1=X
Letter2=-
```

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

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

```
[DriveLetters]
Letter1=-
Letter2=X
```

Why? For each drive the configured letters are checked and the first available one is used. "No letter" is always available, so Letter2 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 criterions

To apply a section to certain drives only, you can define criterions in this section. If you use multiple sections then they must be numbered. The section are tested from `[DriveLetters1]` to `[DriveLetters99]` and finally`[DriveLetters]` (without a number). So, configure special cases at low numbers and more general ones at higher numbers and the most general with the highest number or without a number.

USBDLM 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
[DriveLetters1]
MinDiskSize=10GB
Letter=-

;all others at U, V, W
[DriveLetters2]
Letter1=U
Letter2=V
Letter3=W
```

or in short notation:

```
[DriveLetters2]
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]
Letter1=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.

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 shutted down, then the removal of the folder fails of course. This can be done on next startup or when a user logges on. Configure the folders which contains the NTFS mountpoints.
Sample matching to the DriveLetters section sample above:

```
[DeleteUnusedNtfsMountpoints]
Folder1=C:\_USB
Folder2=
```

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 availlable (the first free letter otherwise) and two NTFS folders:

```
[DriveLetters10]
MaxMountPoints=3
Letter1=U
Letter2=C:\_USB\%DevName%
Letter3=C:\_Share\Drive_%DriveLetter%
```

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

```
[DriveLetters10]
MaxMountPoints=3
Letter1=U,-
Letter2=C:\_USB\%DevName%
Letter3=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 of 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.
So 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, so you have no idea which drive is which there (unless USBDLM makes readable names there)

Conclusion: Do not use NTFS folders as mount points without a good reason.

To get rid of cardreader's drive letters, use USBDLM's No Media No Letter feature.

Settings

Even the section `[Settings]` is mentioned 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 resmue from standyby 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:

```
[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.

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 XP 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 policiy. 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.

USBDLM 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.

External mount events

When drive letters are changed by another software then a service as USBDLM is not notified. To get notified a service need an invisible Window. Under Vista and under XP when more than one user is logged on a window is not good enough, a windows in the context of the interactive user is required. That's what the USBDLM_usr.exe is for. It recieves the notification and sends it down to the USBDLM service.
The OnArrival event and the BalloonTips for mounted TrueCrypt drives depend on this too.

If this is all not required:

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

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 SysInternals DebugView

To activate the log file set WriteLogFile=1, for the live output WriteDebugInfo=1.
To control how detailed the output is, set the LogLevel=1 to 5, 3 is usually good enough.
The file is _USBDLM.LOG in the root folder of the Windows drive by default but can be changed:

```
[Settings]
WriteDebugInfo=1
LogLevel=3
WriteLogFile=1
LogFile=C:\USBDLM.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 service an existing USBDLM.LOG is renamed into USBDLM_BAK.LOG, an existing USBDLM_BAK.LOG is deleted.

Balloon Tips

USBDLM can show a balloon tip with information about the assigned mountpoints of newly attached drives.

```
[BalloonTips]
```

```
Enabled=1
Timeout=6000
```

BalloonTipTimeout 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.

USBDLM includes text resources for some languages. You can change some or all text snippets in the USBDLM.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
```

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:\

The Windows ANSI charset is supported only, no unicode! The maximum number of chars is 47.

If in your language "mounted to" differs in singular and plural, then you can use the additional item `Text_mounted_to_plr`.

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 and file system:

```
Text_singledrive=mounted to %Root%    "%VolumeLabel%" [%FsName%]
```

%Root%, "%VolumeLabel%" and %FsName% are USBDLM variables

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 space 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).

With `NoMediaNoLetter=1` a BalloonTip 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.

Charset

If the Windows default charset is not good enough then you can choose another one.
Sample for for Russian:

```
[BalloonTips]
Text_charset=204
```

Further values defined in the Mircosoft `WinGDI.h:`

GREEK_CHARSET	161
TURKISH_CHARSET	162
HEBREW_CHARSET	177
ARABIC_CHARSET	178
EASTEUROPE_CHARSET	238
RUSSIAN_CHARSET	204
BALTIC_CHARSET	186

Since Vista with an active theme the charset has no effect, the standard Windows charset is used.

Right to left mode text as used for arabic and hebrew isn't supported yet.

Multiple mountpoints

Since V4.3 USBDLM can deal with multiple mountpoints. By default the balloontips 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 off.

Round the drive size

```
[BalloonTips]
RoundSize=1
```

Makes USBDLM show fixed sizes as 256 MB or 512 MB instead of values like 262 MB or 508 MB.
Default is on.

Wait for the last volume

When a device with multiple drives is attached as an USB hard drive with multiple partitions or and USB multislot cardreader, USBDLM can either wait until all volumes are processed or update the balloontip for each volume.
Default is off.

```
[BalloonTips]
WaitForLastVolume=1
```

Suppress Windows Balloons

While an external drive is attached for the first time Windows shows more or less useful balloontips. USBDLM can suppress these balloons if required to show its own. Default is on.

```
[BalloonTips]
SuppressWindowsBalloons=1
```

No Balloontips for TrueCrypt volumes

Since V4.3 USBDLM shows BalloonTips for mounted TrueCrypt volumes. If it shall not do so:

```
[BalloonTips]
ShowTrueCrypt=0
```

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.

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 drives by means of the Window AutoPlay

facility for CD-ROM drives. Very odd solution.
USBDLM 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]
DeviceID1=USB\VID_057C&PID_62FF
DeviceID2=selfinstall
```

The ID has to be eight or more characters or using Wildcards. You can use ListUsbDrives to get the string, see [Letters by DeviceID]

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
[OnBalloonClick1]
open=explorer %drive%

;on right click, open a foto software
[OnBalloonRClick1]
open=c:\program files\fotosoft\fotosoft.exe %drive%
```

You can define several events depending on criterions as shown for [AutoRun].

When more than one drive is shown in the balloon, then the event works for the first one which has a drive letter only.

Do not use `explorer /e` (which shows an Explorer windows with a drive tree) 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...

Migration from V3.x

- Instead of using for each criterion its own section type now there is only one section type, [DriveLettersX]. In each section any criterion, even multiple criterions can be used.
- The order of evaluation isn't fixed anymore, it depends on the numbering only
- USBDLM.INI on the attached drive is discontinued but can be easily emulated, see Configuration.
- RunOnArrival is discontinued but can be emulated by means of the new AutoRun facility
- [DriveLettersMultiSlotCardReader] is discontinued, multislot cardreaders are handled now as every other removable USB drive. A special handling can be achieved thru the DeviceType.
- All sections are good for USB drives only. For other types of drives, configure a BusType.

Samples for the migration

```
Old (V3.x)                              New (V4.0+)
----------------------------------------------------------------------------
```

```
USBDLM.INI on new drive                ;put this in the global USBDLM.INI
                                       [DriveLetters1]
                                       FileExists=%drive%\usbdlm.ini
                                       Letters=%drive%\usbdlm.ini


[MultiSlotCardReaders]                 [DriveLetters5]
BaseFolder=C:\CardReader               DeviceType=MSCR
                                       Letter1=C:\CardReader\%DevName%


[DriveLettersSize1]                    [DriveLetters10]
MinSizeMB=10000                        MinVolumeSize=10GB
Letter1=X                              Letter1=X


[DriveLettersFirewire]                 [DriveLetters20]
Letter1=X                              BusType=Firewire
                                       Letter1=X


[DriveLettersCDROM]                    [DriveLetters25]
Letter1=X                              DriveType=CDROM
                                       Letter1=X


[DriveLettersU3AutoRun]                [DriveLetters30]
Letter1=X                              DriveType=CDROM
                                       DeviceType=U3
                                       Letter1=X

;an USB drive:
[DriveLettersDeviceID1]                [DriveLetters40]
DeviceID1=Corsair Flash Voyager        DeviceID1=Corsair Flash
Voyager
Letter1=X                              Letter1=X

;an ATA drive:
[DriveLettersDeviceID1]                [DriveLetters45]
DeviceID1=IC35L120AVV207-0             DeviceID1=IC35L120AVV207-0
Letter1=X                              BusType=ATA
                                       Letter1=X

[DriveLettersUsbPort1]                 [DriveLetters50]
PortName=3-1-1                         PortName=3-1-1
Letter1=X                              Letter1=X


[Settings]                             [BalloonTips]
ShowBalloonTips                        Enabled
BalloonTipTimeout                      Timeout
ShowSizeInBalloons                     ShowSize
BalloonWaitForLastVolume               WaitForLastVolume
SuppressWindowsBalloons                SuppressWindowsBalloons

[NoBallonsFor]                         [BalloonTipsExclude]

[Settings]                             [Settings]
AutoRunEnabled=1                       AutoRunInf=1


EjectServer                            discontinued
```

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!

Alternatively you can 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\USBDLM_share\usbdlm.ini
```

For this USBDLM.INI only section `[NetworkLetters]` is read!

The file is read from the context 'Local System' of the computer which the USBDLM service is running on with null 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 no 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 USBDLM service only.

Letters1 to 26 can be used in addition.

New Default Letters

In section `[DriveLetters]` you can configure new default letters for USB drives. 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 an USBDLM variable which is replaced 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 USBDLM read drive letters from the USBDLM.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 USBDLM.INI on the drive is present we need the FileExists criterion:

```
[DriveLetters10]
FileExists=%drive%\usbdlm.ini
Letters=%drive%\usbdlm.ini
```

No Media No Letter

The typical 20 in 1 card reader takes for each slot one drive letter, if you have a card of this type or not. No one knows which drive letter is assigned to which slot...
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.

Under Windows 2000 the required notification messages do not exist. Since V4.2 USBDLM can check once a second for a media. This consumes some CPU time, about some seconds per hour, depending on the CPU speed and the number of watched drives.

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 `[NoMediaNoLetter`**`In`**`clude]` (see below).

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

Too discover which process is changing the drive letters you can use the SysInternals RegMon:
`http://technet.microsoft.com/en-us/sysinternals/bb896652.aspx`

Set the filters like this

```
Include: DosDev
```

```
Exclude: Enum
```

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

```
[DriveLetters10]
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 USBDLM, 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 previleges. 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 assingned to USB drives without a media present.

Exclude devices from "no media no letter" function

To make the "NoMediaNoLetter" function work, the USBDLM 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 USBDLM get flooded by wrong removal messages.

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

Sample to exclude the Apple iPod:

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

This is already build in because the iPod is a known trouble device...

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 ListUsbDrives 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.

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 USBDLM assings the first free letter. If you want a certain letter, it must be configured like this:

```
[DriveLetters10]
DeviceID=IDE\DISKIOMEGA_ZIP_100
Letter=Z

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

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 `[DriveLetters]` (without a number).

The first section whose criteria fits to the drive is used. So, configure special cases at low numbers and more general ones at higher numbers and the most general with the highest number of without a number.
I suggest to use the number in steps of 10, so you can later easily insert sections.

The section and value numbers have no relation to anything. Section and values are just evaluated from low numbers upwards to high numbers, that's all.

The ListUsbDrives tool shows for each drives which section is used for then, so you can easily check I you have configured right...

Defauld criteria are:

```
BusType=USB
```

and

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

This means that a section without criteria is appropriate for all USB drives but not USB floppy drives and USB CDROM drives.
In contrast to USBDLM V3.x multislot cardreaders are not excluded here.

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**:

```
[DriveLetters10]
DeviceID1=USB\VID_067B&PID_2517
DeviceID2=USB\VID_1234&PID_5678
Letter1=X
Letter2=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 `Letter3=-` can be configured.

There is no relation between the numbers!

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

```
[DriveLetters10]
DeviceID=USB\VID_067B&PID_2517
Letter=X

[DriveLetters20]
DeviceID=USB\VID_1234&PID_5678
Letter=Y
```

When only one criteria and letter is used then the number can be ommitted.

More samples:

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

```
;FireWire at F:
[DriveLetters10]
BusType=FireWire
Letter=F

;USB >10GB at U:
[DriveLetters20]
MinDiskSize=10GB
```

```
Letter1=U

;all other USB drives at W: or Z:
[DriveLetters]
Letter1=W
Letter2=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:
[DriveLetters10]
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 with numbers. 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:

```
[DriveLetters10]
BusType1=USB
BusType2=FireWire
MinDiskSize=10GB
Letter=F
```

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

```
[DriveLetters10]
BusType=ATA
DeviceID=PCI\VEN_1217&DEV_7135
Letter=P
```

The real device ID is shown by ListUsbDrives (for ATA drives the "Ctrl2 Dev ID", otherwise the "Ctrl Dev ID").

This way it does not work:

```
[DriveLetters20]
DriveType=fixed
Letters=K,L

;special USB harddrive at I:
[DriveLetters30]
DriveType=fixed
DeviceID1=USBSTOR\DISK&VEN_WDC_WD20&PROD_00BB-00GUA0
Letters=I
```

Why? Section 20 'catches' all USB harddrives, section 30 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 (*)`	Presense of a file or folder
`FileNotExists (*)`	Absense of a file or folder
`DirExists (*)`	Presense 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

(*) may cause slower arrivals because the file system must be mounted
(**) since V4.4 no more delays, can be used without hassle

Hint: Fictitious criteria do not work.

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
MinDiskSize
MaxVolumeSize
MaxDiskSize
```

Sample:

```
;USB drives above 40GB at U:
```

```
[DriveLetters10]
MinDiskSize=40GB
Letter1=U

;USB drives with including 10GB and more at X:
[DriveLetters20]
MinVolumeSize=9GB
Letter1=X
```

Using the volume size may cause slower arrivals because the file system must be mounted to read it.

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:

```
[DriveLetters10]
DeviceID1=USB\VID_067B&PID_2517
DeviceID2=USB\VID_1234&PID_5678
Letter1=U

[DriveLetters20]
DeviceID1=USB\VID_9876&PID_5432
Letter1=X
```

The "Friendly Name" as shown by the ListUsbDrives 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.

```
[DriveLetters10]
DeviceID1=Corsair Flash Voyager
Letter1=V
```

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

```
[DriveLetters10]
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 ListUsbDrives tool. The ListUsbDrives_To_Notepad command script redirects its output to a file and opens it in the Windows Editor.

As Device-ID you can use the `'Drive DevID'`, `'Ctrl DevID'` or the `'Friendly Name'`.

An at least 8 characters long part from start is enough. Or use Wildcards.

```
MountPoint     = U:\
Volume Label   = Voyager_2GB
Size           = 2.0 GB (NTFS)
Volume Name    = \\?\Volume{d9e95680-6d80-11db-afb8-000102b35cc3}\
Drive DevID    = USBSTOR\DISK&VEN_CORSAIR&PROD_FLASH_VOY...
Ctrl  DevID    = USB\VID_067B&PID_2517\6&12115AD4&2&1
Ctrl2 DevID    = USB\VID_067B&PID_2515\5&1BBE8508&0&1
Volume DevName = \Device\Harddisk4\DP(1)0-0+25
Disk DevName   = \Device\Harddisk4\DR39
Device Number  = 4
Friendly Name  = Corsair Flash Voyager
USB Version    = 2.0 (high speed)
USB Serial     = ---
USB Port Name  = 5-1-1
```

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

```
[DriveLetters10]
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 device ID was 'USB\VID_067B&PID_2517\6**&**12115AD4**&**2&1'.
But this is one of the devices without a serial. The appearance of the fat & char means that this is a generated number, not the USB serial. Windows generates a unique number on each USB port. So, when using the device ID, omit the last part to make it work on all USB ports:

```
DeviceID1=USB\VID_067B&PID_2517
```

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

```
[DriveLetters10]
DeviceID1=1234567890ABCD
Letter1=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]
DriveType1=floppy
DriveType2=cdrom
DriveType3=removable
DriveType4=fixed
Letter=--
```

Or short:

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

Drive Letters by USB Port

Since V4.4 this can be used without hassle. No more delays. ListUsbDrives 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 (active extension cables or embedded hubs an a very few flash drives) are no more ignored.

It's a bit hard to configure...

```
[DriveLetters10]
PortName=3-2
Letters=U,R

;for multiple ports as the ports of an hub connected to 5-1
[DriveLetters20]
PortName1=5-1-1
PortName2=5-1-2
PortName3=5-1-3
PortName4=5-1-4
Letters=X
```

Or short using Wildcards:

```
;for multiple ports as the ports of an hub connected to 5-1
[DriveLetters20]
PortName1=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 ListUsbDrives tool. ListUsbDrives will show something like this for each USB drive:

```
MountPoint   = U:\
Disk Name    = VOYAGER1GB
Size         = 991 MB
Drive Type   = removable
DOS Device   = \Device\Harddisk8\DP(1)0-0+17
Device Name  = Corsair Flash Voyager
USB Version  = 2.0 (high speed)
Serial       = ---
USB Port     = 3-2
```

For this drive 3-2 is the setting for "PortName" we need. If there is an 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.
With USB 2.0 the physical USB ports have two incarnations - one for USB 1.1 and one for USB 2.0.
Run Microsoft UVCView to get a tree view of your USB ports and devices to see what I mean.
There is no safe way to determine which USB 1.1 incarnations shares its physical ports with which USB 2.0 incarnation. So the only way to get the same letter for USB 1.1 and USB 2.0 drives on the same port is to run ListUsbDrives twice - one time with an USB 1.1 drive attached and one time with an USB 2.0 drive attached. But this is true for ports of root hubs only. Ports of normal USB hubs have one incarnation only.

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.
Sample:

```
MountPoint      = U:\
Disk Name       = VOYAGER1GB
Size            = 991 MB
Drive Type      = removable
DOS Device      = \Device\Harddisk8\DP(1)0-0+17
Device Name     = Corsair Flash Voyager
Host Ctrl DevID = PCI\VEN_8086&DEV_2659&SUBSYS_3005103C&REV_03\3&B1BFB68&0&E9
Host Ctrl Name  = Intel(R) 82801FB/FBM USB Universal Host Controller - 2659
USB Version     = 2.0 (high speed)
Serial          = ---
USB Port        = 3-2
```

The resulting "long" port name whoud be here PCI\VEN_8086&DEV_2659&SUBSYS_3005103C&REV_03\3&B1BFB68&0&E9-2.
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:

```
[DriveLetters20]
PortName1=PCI\VEN_8086&DEV_2659*-2
Letters=X
```

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:

```
[DriveLetters10]
VolumeLabel1=Monday
VolumeLabel2=Wednesday
VolumeLabel3=Friday
Letter1=U

[DriveLetters20]
VolumeLabel1=Tuesday
VolumeLabel2=Thursday
VolumeLabel3=Saturday
Letter1=X
```

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

You can use Wildcards:

```
[DriveLetters10]
VolumeLabel1=Backup?
Letter1=U
```

This would fit for a VolumeLabels Backup1, Backup2 etc.

Using this may cause slower arrivals because the file system must be mounted to read the volume label.

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.

Determine the serial:

Start -> Run
CMD

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

The ListUsbDrives tool shows the serial too.

Sample:

```
[DriveLetters10]
VolumeSerial=B92B-8000
Letter=U
```

Using this may cause slower arrivals because the file system must be mounted to read the volume serial.

The volume serial can be changed by means of the commandline tool VolumeID from SysInternals.

Wildcards can be used:

```
[DriveLetters10]
VolumeSerial=B92B-80??
Letter=U
```

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 drives
`UNKNOWN`	unknown, some card readers in notebooks
`RAMDISK`	RAM drives (many RAM drives appear as FIXED, check with ListUsbDrives)
`ANY / ALL`	all types (**use it carefully!**)

Sample for USB-CD-ROMs at R:

```
[DriveLetters10]
DriveType=CDROM
Letters=R
```

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

```
[DriveLetters10]
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:

```
[DriveLetters10]
UserGroup=Administrators
Letters=X,Y,Z

;no USB drives for all others
[DriveLetters90]
DriveType=ALL
Letter=--
```

Do not configure like this:

```
;all others
[DriveLetters90]
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
- `TrueCrypt` -> a drive that is used as TrueCrypt container (or as container for any other encryption software)
- `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]` events, not for drive letters!)
- `BitLocker` -> a BitLocker volume
- `AllZero` -> first sector contains zeros only (unformatted or non Windows drives)
- `UnrecognizedVolume` -> Windows says ERROR_UNRECOGNIZED_VOLUME, e.g. Ext3 drive

The ListUsbDrives tool shows the types.

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

```
[DriveLetters10]
DeviceType=MSCR
Letters=L,M,N,O
```

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

```
[DriveLetters10]
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 junk 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:

```
[DriveLetters10]
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:

```
[DriveLetters10]
DeviceType=TrueCrypt
Letter1=-
```

Sample to hide U3 AutoRun CD-ROMs:

```
[DriveLetters10]
DriveType=CDROM
DeviceType=U3
Letter1=-
```

U3 is a standard fur 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 an 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 irreversiblely then use the U3 uninstall tool:
http://u3.com/uninstall

USBDLM 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... USBDLM does not recognize such drives as fake CD-ROMs so far.

Drive Letters by Bus Type

Bus Typ means the way a drive is attached to the computer, like USB, FireWire, SCSI ect.

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 for non USB drives.

ListUsbDrives shows the BusType of all drives if it's started with parameter -a (like all). That's what the ListUsbDrives_To_Notepad.cmd does.

The BusTypes are:

`USB`
`FireWire`
`SCSI`
`ATAPI`
`ATA`
`SSA`
`Fibre`
`RAID`
`iSCSI`
`SATA`
`SAS`
`SD` (SD cardreader)
`MMC` (MMC cardreader)
`Unknown` (internal notebook cardreaders, virtual drives, legacy floppies)
`ANY or ALL` (any BusType - **use it carefully, it might remount your internal drives!**)

Under XP SATA drives never have the BusType=SATA, it's not defined there. They will have ATA or SCSI as BusType, sometimes RAID... Even under Vista I've never seen SATA so far. ListUsbDrives 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      = [DriveLetters10], [OnArrival20]
Volume DevID      = STORAGE\VOLUME\1&30A96598&0&SIGNATURE379DB9F7OFFSET7E0000LENGTH2E933D8200
Drive DevID       = SCSI\DISK&VEN_&PROD_ST3200822AS&REV_.320\5&279143B0&0&000000
Ctrl  DevID       = PCI\VEN_197B&DEV_2360&SUBSYS_2360197B&REV_02\4&F39E1D8&0&00E0
Host Controller   = JMicron JMB36X Controller
Friendly Name     = ST3200822AS SCSI Disk Device
```

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. Befor letting USBDLM work with the INI let ListUsbDrives show which sections are used for the drives, see items "`INI Sections`".

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

```
[DriveLetters10]
BusType1=USB
BusType2=FireWire
MinDiskSize=100GB
Letter1=X

;alternative notation with comma
[DriveLetters10]
BusTypes=USB,FireWire
MinDiskSize=100GB
Letter1=X
```

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

```
[DriveLetters10]
BusType=ATA
DeviceID1=IC35L120AVV207-0
Letter1=X
```

Sample for internal laptop cardreader with BusType SCSI:

```
[DriveLetters30]
BusType=SCSI
DeviceID1=SCSI\DISK&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:

```
[DriveLetters30]
BusType1=SD
BusType2=MMC
Letter1=X
```

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

```
[DriveLetters10]
BusType=SCSI
DeviceID1=ELBY CLONEDRIVE
DeviceID2=SCSI\CDROM&VEN_GENERIC&PROD_DVD-ROM
DeviceID3=SCSI\CDROM&VEN_DAEMON
Letters=V-Y
```

Drive Letters by Partition Number

Logical drives on Harddisks have a partition number, the ListUsbDrives tool shows them.

Sample for hiding partition number 2 on an USB harddrive with DeviceID `USB\VID_04B4&PID_6830\1234567890AB`:

```
[DriveLetters10]
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.

Letter by Existence of a File or a Folder

```
[DriveLetters10]
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:

```
[DriveLetters10]
BusType1=USB
BusType2=FireWire
BusType3=SCSI
FileExists=%Drive%\drive_a.id
Letter1=A
```

Since V4.2 you can use wildcards. The evaluation is done by Windows, therefore it's different form the USBDLM 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:

```
[OnArrival10]
FileExists1=%Drive%\*.jpg
FileExists2=%Drive%\DCIM
open="%ProgramFiles%\ImageViewer\ImageViewer.exe" %Root%

[OnArrival20]
FileExists1=%Drive%\*.avi
FileExists2=%Drive%\*.mpg
FileExists3=%Drive%\*.mpeg
FileExists4=%Drive%\*.mkv
FileExists5=%Drive%\*.mov
open="%ProgramFiles%\VideoPlayer\VideoPlayer.exe" %Root%

[OnArrival25]
DirExists1=%Drive%\VIDEO_TS
open="%ProgramFiles%\DvdPlayer\DvdPlayer.exe" %Root%
```

Since V4.3 USBDLM can test for absence of a file (`FileNotExists`).
Sample to mount a TrueCrypt volume on X: if X: is available:

```
[OnArrival10]
DeviceType=TrueCrypt
FileNotExists=X:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /v %PartitionName% /lX /q
```

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

```
[OnArrival10]
DeviceType=TrueCrypt
DirNotExists=X:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /v %PartitionName% /lX /q
```

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

Drive Letters by active User

The active user can be configured using its user name or the name of its local group.

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. This is done immediately after logon so the user has no chance to break in the communication between service and user instance. Under Windows 2000 it's more critical because here the USBDLM service has to wait for the explorer.exe, so an very advanced user might be able to breach the communication.
Use server groups only if required.

To activate reading server groups set

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

Sample for the user name:

```
;drive letter for the parents
[DriveLetters10]
UserName1=Homer
UserName2=Marge
Letter1=X
Letter2=Y

;no drives for the kids
[DriveLetters20]
UserName1=Bart
UserName2=Lisa
UserName3=Maggie
DriveType=ANY
Letter1=-
```

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

Sample for group names:

```
[DriveLetters10]
UserGroup=Administrators
Letter1=X
Letter2=Y

;no USB drives for all others
[DriveLetters20]
Letter1=-
```

Or reverse:

```
;no drives for normal users
[DriveLetters10]
UserGroup1=Users
UserGroup2=Guests
Letter1=-

;all other users can
[DriveLetters20]
Letter1=X
Letter2=Y
```

Or using the user's admin state

```
[DriveLetters10]
UserIsAdmin=1
Letter1=X
Letter2=Y

;no drives for non admins
[DriveLetters20]
UserIsAdmin=0
DriveType=ALL
Letter1=-
```

Under Vista 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,
see Letters by Device ID.

```
;this drive always works because section 15 is tested before section 20
[DriveLetters15]
DeviceID=USB\VID_067B&PID_2517
Letter1=X
Letter2=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.

Priorities

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

USBDLM 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 USBDLM remounts it to the next available letter.
No need no 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, USBDLM 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 TweakUI 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.

USBDLM's AutoRun functions

USBDLM 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 commandline is not executed.

In contrast to Windows' AutoRun this 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.

USBDLM supports only the starting of executable files (EXE, COM, BAT, CMD, LNK etc), document files can work but there is no guarantee. E.g.
```
open=test.txt
```
may not work. Use then something like
```
open=notepad test.txt
```

instead.

If the path contains spaces and commandline 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

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.

Executable files are searched first in the working directory, then in the Windows search path (environment variable PATH). An additional search path can be configured, e.g.

```
[Settings]
OpenSearchPath=C:\BatchFiles
```

AutoRun events on insertion or removal of a cardreaders's media configured for the first time may work after reattaching the device or after a restart of the USBDLM service only. This is because for receiving these notifications USBDLM 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 USBDLM's startup.

The USBDLM 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]
UsbdlmVariablesToOpenEnvironment=%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 USBDLM 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 commandline when a drive is attached. With each Windows version this has been limited more an more. Under XP this works with CD-ROM drives only whithout further user inquiry. This is the reason for such great solutions as U3 drives or selfinstaller devices which are using a fake CD-ROM drive.
Under Vista even this works after asking the user only.

USBDLM 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

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=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 USBDLM 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

Under 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 can be added in milliseconds. The maximum value is 60000:

```
delay=2000
```

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

```
AutoRunKey=MySecretKey
```

Only if the same line is found in the autorun.inf 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

```
[OnArrival10]
open=calc.exe
```

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

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

```
[OnArrival10]
open=calc.exe
restricted=1
```

or with full system privileges in the context 'local system':

```
[OnArrival10]
open=calc.exe
system=1
```

You will never see this program because it's in the "Local System" 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 as splitted 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":

```
[OnArrival10]
open=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!
Furthermore there is the default criteria `MinDiskSize=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 `MinDiskSize=-`

USBDLM 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=calc.exe
OneInstance=1
```

To ensure the integrity of the executable file, an MD5 hash value can be configured:

```
[OnArrival]
open=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 doesn't matter.

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)

```
[OnArrival10]
VolumeLabel1=CANON_DC
VolumeLabel2=NIKON_DC
open="C:\Program Files\PhotoSoftware\PhotoSoftware.exe" %root%

[OnArrival20]
Letter=X
open=

;all others
[OnArrival30]
open=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=cmd /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:

```
[OnArrival10]
FileExists=%drive%\DATA.TXT
open=cmd /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

```
[OnArrival10]
FileExists=%drive%\secret.tc
open="C:\Program Files\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\secret.tc" /l T

[OnArrival11]
DeviceType=TrueCryptVolume
open=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
[DriveLetters10]
DeviceType1=TrueCrypt
DeviceType2=ReadSharingViolation
Letter=-

;mount it on J:
[OnArrival10]
DeviceType=TrueCrypt
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v %PartitionName% /l J

;an Explorer window with the mounted TrueCrypt volume
[OnArrival11]
DeviceType=TrueCryptVolume
open=explorer %root%
```

`%PartitionName%` is an USBDLM variable which USBDLM 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 USBDLM cannot read test data - it's rejected with ERROR_SHARING_VIOLATION, that's what the name comes from.
This usually happens when the USBDLM 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 USBDLM 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.

Sample 5:
- An USB drive on U: and create a share `Drive_U` for it, on "safe removal" delete the share:

```
[DriveLetters10]
Letter1=U

[OnArrival10]
Letter=U
open=net share FlashDrive_U=U:\

[OnRemovalRequest10]
Letter=U
open=net share FlashDrive_U /D
```

Sample 6:

- An USB flash drive at A:, if A: is in use, remout A: to B: and assign A: then:

```
;mount to A: if available, otherwise unmount
[DriveLetters10]
Letter1=A
Letter2=-

;if unmounted, remount A: to B: and assign A:
[OnArrival10]
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 an USBDLM variable which USBDLM replaces by something like `\\?\Volume{aa6d706a-15da-11dc-a38f-0013d31dd4c5}\`, as expected by the Windows commandline tool `MountVol`.
`ReMount` is my commandline tool for changing drive letters. Download:
http://www.uwe-sieber.de/files/remount.zip

Assign A: to the previous drive when the USB flash drive has been removed:

```
[OnRemoval10]
Letter=A
open=ReMount B: A:
system=1
```

The line `system=1` makes USBDLM executing the command line in the context 'local system' 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.
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:
- 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
```

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 is closes after 10 Seconds. By clicking on the countdown the timeout starts over again. By commandline **-**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).

AutoRun on and after Removal

In analogy to the OnArrival function USBDLM can execute a commandline when a drive is "prepared for safe removal" and after a drive has been removed.

1. On preparation for safe removal

When an 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 under Vista. 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:

```
[OnRemovalRequest10]
open=cmd /c copy "C:\test.txt" %drive%\backup
```

Same, but for drive X: only:

```
[OnRemovalRequest20]
Letter=X:
open=cmd /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:

```
[OnRemovalRequest30]
FileExists=%drive%\backup\test.txt
open=cmd /c copy "C:\test.txt" %drive%\backup
```

Unmount a TrueCrypt volume T: when the file secret.tc is found on the drive to remove:

```
[OnRemovalRequest40]
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 save removal by means of my commandline 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:

```
[OnRemovalRequest10]
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:

```
[OnRemovalRequest20]
Letter=X
CloseWindowText=Calc
CloseWindowClass=SciCalc
afterdelay=2000
```

Wildcards can be used when the windows text depends on the opened document:

```
[OnRemovalRequest30]
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:

```
[OnRemovalFailed10]
Letter=U
open=net share FlashDrive_U=U:\
```

3. After removal

Sample: Drive A: has been removed, another drive shall be mounted to A:

```
[OnRemoval1]
Letter=A
open=mountvol A: \\?\Volume{433619ea-c6ea-11d9-a3b2-806d6172696f}\
system=1
```

MountVol is a commandline tool which comes with Windows 2000 and higher.
`system=1` makes USBDLM execute it in the local system context to give it the required privileges in case the user isn't an admin. It's invisible then.

Limitation

All these removal events are triggered only if the volume was present for at least 5 Seconds.

AutoRun on System Events

In analogy to the AutoRun function, USBDLM can execute a commandline on system events.

There are no criteria evaluated.

The events are

OnServiceStart
OnServiceStarted
OnServiceStop
OnServiceShutdown
OnUserLogon
OnUserLogoff
OnUserLock
OnUserUnlock
OnUserConnect
OnUserDisconnect
OnSleepRequest

OnSleep
OnResume

The names are (hopefully) self explaining.

OnUserConnect and OnUserDisconnect mean the connection of the input devices (mouse+keyboard) with the desktop. They are triggered for instance when the user is switched by means of the Fast User Switching.

OnSleepRequest and OnSleep does not guarantee that something is executed (and finished) before the system goes to sleep. OnSleep is often triggered on wake up...
OnSleepRequest does not exist under Vista and Windows 7.

The OnUserXxxx events do not work under Windows 2000.

Sample for creating a network drive on user logon:

```
[OnUserLogon]
open=cmd /c net user X: \\server\share /user:username password
openstyle=min
```

TrueCrypt

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.

http://www.truecrypt.org

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 than random data.

Only the TrueCrypt application and its driver are able to mount the container and make its contents available decrypted under a new drive letter.

Partition as Container

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.

It's 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 this sticks for exactly this drive 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 partition 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 USBDLM service is restarted while the TrueCrypt container is mounted, then we want the letter for the container partition removed but the volume shall not tried no mount again by TrueCrypt, that's what this sample does:

```
;remove the volume's drive letter, it's useless
[DriveLetters10]
DeviceType1=TrueCrypt
DeviceType2=ReadSharingViolation
Letter=-

;mount it on J:
[OnArrival20]
DeviceType=TrueCrypt
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v %PartitionName% /letter J

;an Explorer window with the mounted TrueCrypt volume
[OnArrival11]
DeviceType=TrueCryptVolume
open=explorer %root%
```

`%PartitionName%` is an USBDLM variable which USBDLM replaces with something like `\Device\Harddisk2\Partition1` as expected by TrueCrypt.

File as Container

When a drive arrives then USBDLM can check for the presence of a file and let TrueCrypt mount it:

```
[OnArrival30]
FileExists=%drive%\secret.tc
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\secret.tc" /letter J
```

Better check if the target letter is available by means of the `DirNotExists` criteria:

```
[OnArrival30]
FileExists=%drive%\secret.tc
DirNotExists=J:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\secret.tc" /letter J
```

The TrueCrypt Volume

When TrueCrypt mounts a container, then it makes its contents available under a drive letter. This drive isn't a full blown "storage volume", it is implemented partially only.
USBDLM sees the arrival of the drive letter and can trigger an OnArrival for it. The TrueCrypt drive's DeviceType is "TrueCryptVolume". That's all. USBDLM knows it's drive letter and makes it available in the USBDLM variables as %drive%. But it cannot deal with the drive's drive letter. In fact it could but TrueCrypt would loose track.

By means of the Windows tool MOUNTVOL the TrueCrypt volume can be mounted into an existing folder on an NTFS drive:

```
[OnArrival11]
DeviceType=TrueCryptVolume
open1=mountvol "C:\TrueCrypt" %VolumeName%
system1=1
open2=explorer %root%
```

USBDLM 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, USBDLM can let unmount a TrueCrypt volume. Unfortunately TrueCrypt can unmount by drive letter only.

```
[OnRemovalRequest10]
FileExists=%drive%\secret.tc
DirExists=J:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dismount J
```

The parameter `/q` or `/quit` is important because USBDLM answers the removal request not before the started TrueCrypt.exe ends. Without `/quit` TrueCrypt shows a window and USBDLM waits and waits... Finaly Windows will say "The service USBDLM prevents the removal".

BitLocker To Go

BitLocker is a full storage volume encryption introduced with Windows Vista. But BitLocker supports local disks only.

With Windows 7 we get "BitLocker To Go" which allows to encrypt hotplug drives. It comes with Ultimate and Enterprise Edition only.

BitLocker works with a "Filter Driver" which blocks or grants access to the volume.

A hotplug volume encrypted wiht BitLocker comes in two different shapes. It depends on the drive's file system which one we get.

BitLocker encrypted FAT/FAT32 drives still have a 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, even Windows 7. Seems there was a communication deficit between the developer teams at Microsoft.

What USBDLM can do...

1. On Windows with support for "BitLocker To Go"

Encrypted volumes are detected as such, they get the DeviceType `BitLocker`.

The BalloonTip is shown immediately. AutoRun events are triggered when the 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

```
[Driveletters80]
DeviceType=BitLocker
Letter=-
```

Furthermore we can show a manually made BalloonTip which informs the user about the problem:

```
[OnArrival30]
DeviceType=BitLocker
open=%usbdlmpath%\usbdlm.exe -balloon -time=20000 -title="%FriendlyName%"
-text1="Cannot read BitLocker encrypted volume" -icon=110
```

"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 presense of the BitLockerToGo.exe and execute it:

```
[OnArrival20]
FileExists=%drive%\BitLockerToGo.exe
open=%drive%\BitLockerToGo.exe
```

Windows Explorer Icon and Label

USBDLM 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 is does not. It's not USBDLM'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, USBDLM recreates them if required.

USBDLM 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:

```
[Explorer1]
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).

Sample to get a label "USB-Drive" for USB drives:

```
[Explorer2]
BusType=USB
DefaultLabel=USB-Drive
```

You can use variables here.

Sample for the device name for USB drives which have no volume label:

```
[Explorer2]
VolumeLabel=-
BusType=USB
DefaultLabel=%DevName%
```

Limitations under Windows 2000

Under Windows 2000 there is a limit of 31 characters for the default label.
To take the new settings effect the Explorer may need a View->Refresh or the F5 key.

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.

USBDLM can remove this removable flag from the registry. This hides a drive from the list.

Under **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 USBDLM made the registry changes.

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 an 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 beeing shown under "Safely Remove Hardware" - USBDLM then set the removable flag in the registry:

```
[AddToSafelyRemoveHardware]
DeviceID1=IDE\DISKFUJITSU_MHV2040BH*00000025
```

But even you can drives listed there, the safe removal facility usually fails to remove IDE drives, while my commandline tool RemoveDrive works.

Devices which are not related to drives can be handled too. But because USBDLM stores no data about other devices than drives, the complete device ID must be configured:

```
[HideFromSafelyRemoveHardware]
CompleteDeviceID1=USB\VID_1758&PID_2004\050712014270000083
```

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 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.

Device Description

In the Safe Removal Menu 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. USBDLM 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 USBDLM writes the device's "FriendlyName" to the DeviceDesc value, something like "Corsair Flash

Voyager". You can configure this using USBDLM variables.

Sample for FriendlyName and BusType for all USB drives:

```
[Settings]
SetDeviceDesc=2
DeviceDesc=%FriendlyName% %BusType% Device
```

The result would be something like "Corsair Flash Voyager USB Device"

The value DeviceDesc is changed for the USB device. Since an 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 an 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.

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. If it shall be handled by USBDLM anyway then force it by using a criterion `DeviceID1` in its drive letter section:

```
[DriveLetters10]
DeviceID1=ven_apple&prod_ipod
Letter=X
```

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
%Label%	drive name	My flash drive
%VolumeLabel%	drive name	My flash drive
%DriveType%	drive type	REMOVABLE
%FsName%	file system	FAT
%Size%	volume size	2 GB
%VolumeSize%	volume size	2 GB
%DiskSize%	disk size	20 GB
%DosDevName%	'DOS device name'	\Device\Harddisk3\DP(1)0-0+d
%PartitionName%	Partition name	\Device\Harddisk2\Partition1
%DeviceNumber%	device number	2
%PartitionNumber%	partition number	1
%VolumeName%	volume name	\\?\{GUID}\
%Date%	Date (yymmdd)	090331
%Date4%	Date(yyyymmdd)	20090331
%Time%	Time (hh:mm:ss)	12:00:00
%UserName%	user name	Administrator

These variables can be made available as environment variables for processes started by USBDLM.
Which variables are required must be configured.

Sample:

```
[Settings]
UsbdlmVariablesToOpenEnvironment=DevName, FsName, VolumeLabel
```

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, Bub	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

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 drives is lost when a restore point is created while the drive is no attached. In practice it must be disabled again and again for USB hard drives.

Vista only watches the system drive.

USBDLM 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
```

The folder 'System Volume Information' will be created anyway because it's used for other Windows background stuff.

To prevent at least the system restore scans around the disk and fills the "System Volume Information" folder, USBDLM 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
```

Technical Information

Here some background information, just what came into my mind...

Window Service

USBDLM 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 "Local System" and have highest privileges here. USBDLM is a Win32 service, not a kernel service. A kernel service could have the start type "System" 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.

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 USBDLM shows the balloons directly. Otherwise it starts itself in the user's context and shows the balloon.

You can remove the "interactive" flag, USBDLM will show the balloons then indirectely.

The USBLDM.MSI has the "interacitve" Flag but USBDLM removes it when it is started first under Vista.

TrueCrypt

TrueCrypt volumes are not fully implemented. For instance there is no notification when the volume is mounted. Maybe this is not possible because the volume is made availlable for the current user only.

But even under Windows 7 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.

To receive this USBDLM has an invisible window. But this does not work when the user is running in a different session. USBDLM need a helper application then which receives the message an passes it "down" to the serivce. That's what the USBDLM_usr.exe is for. It is loaded when:

- `ExternalMountEvents=1`
- an `[OnArrival]` with `DeviceType=TrueCryptVolume`
- `[BalloonTips] ShowTrueCrypt=1`
- USBDLM Win32 under Vista x64 with `NoMediaNoLetter=1` (workaround for `Windows-Bug in the WoW64 layer`)

Event Logging

USBDLM can write drive arrival and removal events to the Windows application log.

```
[Settings]
EventLog=1
```

It writes then items like these:

```
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
```

Since V4.3 this can be configured using USBDLM 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%
```

Hint: While the Windows Event Viewer is open, the USBDLM.EXE cannot be deleted or replaced.

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
```

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.

Windows Explorer fails when deleting a folder

This is one of many issues when a drive is mounted into an NTFS folder, it's not an USBDLM issue. Please read section "Mounting to NTFS folders" under configuration.

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 a ethernal 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.

Devices which fail to start under Vista and Windows 7

For removable USB drives Vista installs a WPD file system driver too. WPD is "Windows Portable Device", a programming interface for devices as MP3 players, mobile phones and others. If an USB mass storage device has no drive letter, then the WPD file system driver for this drive fails to start. It's not USBDLM's fault, it's just the missing drive letter.
The device is still working as mass storage device, only WPD is not available which doesn't hurt.
If you don't want this WPD stuff at all then you can deactivate the service "Portable Device Enumerator Service". The Windows Media Player stops then synchronizing with USB mass storage devices.

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 ListUsbDrives only and when the USBDLM service is restarted while a TrueCrypt drive is alread mounted. It is detected then with a DeviceType `ReadSharingViolation` instead of `TrueCrypt`. Read more in section TrueCrypt.

History

V4.4.1 (11. November 2009)

- Bugfix: Hiding drives from the "Safely Remove Hardware" might not work under Vista, under Window 7 it still does and will not work for "Removable" drives
- Bugfix: ListUsbDrives always showed a warning about an 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 on XP

V4.4.0 (1 November 2009)

- Bugfix: BusTypes in list notation (e.g. BusTypes=USB,SCSI) didn't work
- Bugfix: Deactivating System Restore for new drives didn't work
- Bugfix: OnArrival did not work for drives with no media present, even MinDiskSize=- was configured
- Bugfix: ListUsbDrives 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
- New: Alternative notation of the USB Port Name
- New: Letters by USB-Port much faster now
- New: Label and Icon from autorun.inf 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" 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: ListUsbDrives 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: ListUsbDrives shows the sections from the USBDLM.INI which are used for a drive
- New: Faster operation by gathering required drive information only
- New: Multiple NTFS folders can be assigned as mount points, also in addition to a drive letter
- New: In sections [OnArrival], OnRemoval etc up to 10 commands can be executed (open1 to open9 and the known open)
- New: [OnArrival] and AutoRun.inf for TrueCrypt volumes
- New: When a DeviceID is configured then the BusType for non USB drives can be omitted, same for the DriveType e.g. for CDROM drives
- New: [OnArrival] can start a program under Vista with full rights ("elevated")
- New: Criteria UserIsAdmin is under Vista related to the "elevated" user, so it is 1 for an Admin under UAC
- New: Criteria for absence of a file (`FileNotExists`)
- New: [OnRemovalFailed] is executed asynchronousely to avoid event ping pong when using RemoveDrive

- New: AutoRun on System Events
- New: Handling of first time arrivals under Windows 2000 optimized
- New: Using [OnArrival], OnRemoval etc there is the new default criterion MinDiskSize=1, so a media must be present
- New: Drive letters can be checked [on resume] from standby and hibernation mode
- New: Drive letters by Volume Serial Number
- New: Drive description for USB drives in the menu of "Safely Remove Hardware"
- New: Balloons under Vista in almost correct design
- New: Balloons with fade out effect and shadow (if activated in the Windows settings)
- New: Using USBDLM Variables for environment variables can (and should) be configured in detail
- New: Section [ExcludedLetters] renamed to [NetworkLetters]

V4.2.5 (29th March 2009)

- Bugfix: Crippled BalloonTip if quotation mark in device name
- Bugfix: Sometimes wrong size in BalloonTip and ListUsbDrives
- Bugfix: The execution of the autorun.inf 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] failed when the path contained spaces and commandline parameters are used
- Bugfix: The execution of [OnArrival] failed when the arrived drive has no mountpoint and no WorkDir is specified
- Bugfix: Under Windows 2000 the port names of USB1 devices attached to USB2 ports where not determined

V4.2.3 (14th September 2008)

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

V4.2.2 (17th August 2008)

- Bugfix: [OnArrival], [OnRemoval] works again without configuring a workdir when the drive has no mountpoint
- Bugfix: [OnRemoval] works again without configuring a workdir
- Bugfix: [OnRemoval] is triggered on removal of a media from a card reader
- Bugfix: New BusTypes 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 the root of the drive is the working directory again

V4.2.0 (12thApril 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 works under Windows 2000
- New: most settings take effect without restarting the USBDLM service
- New: Wildcards can be used configuring several parameters
- New: drives can be completely ignored
- New: drives and other devices can be removed from or added to the safely remove hardware facility
- New: the policy "Optimized for performance" can be set for USB drives
- New: on AutoRun USBDLM variables can be made available to the executed program as environment variables
- New: on AutoRun 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 works under Vista64
- New: NoMediaNoLetter can be configured more detailed
- New: NoMediaNoLetter can be forced for certain drives by a section [NoMediaNoLetterInclude] (analog to [NoMediaNoLetterExclude])
- New: Drive letters in short notation, like Letters=R,X,Y,Z
- New: User=- as criterion for 'no user logged on'

V4.0.2 (21th August 2007)

- Bugfix: ListUsbDrives 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 addtional criterion Letter=- was configured
- Bugfix: Under Windows 2000 partition numbers and balloontips didn't work correctely 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 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: BalloonTips 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 'Local System' 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 mults-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 are applied now on startup too
- new funktion of section [ExcludedLetters]: These letter will never be used anymore for USB drives
- Read additional [ExcludedLetters] from a central USBDLM.INI
- detects and handles U3 autorun drives
- Execution of the open= line in an autorun.inf on removable drives
- Global AutoRun on arrival of any USB or Firewire drive
- Balloon-Tips 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 an USB drive without media is attached
- New/Fix: service name changed from 'USB Drive Letter Manager' to 'USBDLM'

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 commandline 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 mana**n**ger')
- 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 it 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 USBDLM 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 fixded 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 an USBDLM.INI on the newly attached drive
- Exclude list for volume labels
- No remounting when an usbagent.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 is 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.

A Site Licence allows anyone in the organization making the purchase, within a 160 km (100 mile) radius of the purchasing location, to use the software. This includes people from that organization location who are travelling but it does not include people with locations outside of the 160 km radius.

A World-Wide Licence allows anyone in the organization making the purchase, who is on the planet Earth, to use the software.

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 archives is unchanged. Re-packaging is allowed for internal use.

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/usbdlm_e.zip

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.