Index

USB Drive Letter Manager V5.2.5

Einstieg		
<u> </u>		
T 11		
<u>Installation</u>		

Konfiguration

Probleme

History

Lizenz-Vereinbarung

Homepage:

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

Tipps für USB-Laufwerke:

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

E-Mail:

mail@uwe-sieber.de

Einstieg

Der USB Drive Letter Manager 'USBDLM' ist ein Windows-Dienst, der Kontrolle über die Zuordnung von Laufwerksbuchstaben verschafft.

Man kann Buchstaben für USB-Laufwerke definieren, auch abhängig vom USB-Port, der Laufwerks-Größe

Konflikte mit Netz- und Subst-Laufwerken werden auch ohne Konfiguration behoben.

Nicht-USB-Laufwerke können auch behandelt werden, indem deren <u>BusType</u> oder <u>Geräte-ID</u> konfiguriert wird.

Als Windows-Dienst ist USBDLM, einmal installiert, unabhängig von den Rechten des angemeldeten Nutzers.

USBDLM läuft unter Windows XP, Server 2003, Vista, Windows 7, Server 2008 (R2), Windows 8, Server 2012 (R2).

Unter folgenden Systemen wurde entwickelt und getestet:

- Windows XP SP3
- Windows 7 Ultimate
- Windows 8 Professional x64

Grundlegende Funktionen wurde getestet unter:

- Windows XP x64 SP1
- Windows Vista Ultimate x64 SP2
- Windows 7 Ultimate x64
- Windows 2008 Server R2 x64
- Windows 8.1 Pro

USBDLM ist nicht kompatibel mit Microsoft Multipoint Server Clients.

Wenn von Irgendwas+ die Rede ist, meint das ein Betriebssystem und alle Nachfolger, Vista+ z.B. Vista, Windows 7, Windows 8, 8.1, 9 usw. sowie deren Server-Versionen.

Installation

USBDLM kommt nicht in einer SETUP.EXE, die Installation erfolgt manuell, siehe hier: <u>Installation</u>. Alternativ gibt's noch MSIs - die installieren den Dienst ohne weitere Rückfragen nach %ProgramFiles%\USBDLM.

Konditionen

USBDLM ist Freeware für die private Nutzung sowie für Bildungseinrichtungen. Anderenfalls ist - nach einer 30-tägigen Testzeit - pro Computer eine Lizenz nötig, siehe unten.

Bildungseinrichtungen gelten als solche, wenn Zahl der eingeschriebenen Schüler und Studenten höher ist als die der bezahlten Mitarbeiter.

Preise gibt's hier:

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

Lizenz-Vereinbarung

Die Software ist auch ohne Lizenz voll funktionsfähig - es gibt keine "Vollversion".

Wozu?

Beim erstmaligen Anschließen eines USB-Datenträgers (Stick, CardReader, Festplatte) vergibt Windows dafür den ersten freien lokalen Laufwerks-Buchstaben - der Datenträger wird auf den Buchstaben 'gemountet', dort 'bereitgestellt'. Den Buchstaben kann man dann in der Datenträger-Verwaltung mit einiger Klickerei umstellen und im Normalfall wird Windows sich das für genau diesen Datenträger merken. Windows merkt sich aber nur genau eine Zuordnung pro Buchstabe.

Für jeden neuen USB-Datenträger geht das Ganze von vorne los, ebenso wenn man ein Gerät, das im Widerspruch zu den USB-Standards keine Seriennummer hat, an einen anderen USB-Port anschließt.

Netz- und Subst-Laufwerke sind ab Windows XP nutzerspezifisch und damit keine 'lokalen' mehr. Hat man auf dem ersten freien lokalen Buchstaben ein Netz- oder Subst-Laufwerk, bekommt ein neues USB-Laufwerk unter XP genau diesen Buchstaben und es scheint, als hätte das Laufwerk keinen Buchstaben bekommen. Tatsächlich kommt man nur nicht ran...

Mit dem XP-Servicepack 3 wird dieses Problem für erstmals angeschlossene Laufwerke behoben.

Mit USBDLM kann man

- Konflikte mit Netz- und Subst-Laufwerken beheben lassen (keine Konfiguration nötig)
- Laufwerksbuchstaben für Netz- und Subst-Laufwerke reservieren
- Für alle neuen USB-Datenträger eine Liste von <u>Standard-Buchstaben</u> vorgeben, auch abhängig von einer Vielzahl an <u>Kriterien</u>, z.B. Laufwerks-Typ, Anschluss (USB, FireWire), USB-Port, Datenträgerbezeichnung, Größe des Datenträgers, Hardware-ID u.v.a.
- Laufwerksbuchstaben von Kartenlesern entfernen und genau dann einen Buchstaben zuordnen, wenn eine Karte eingelegt wird
- Den Nutzer vor dem Zuordnen eine Laufwerksbuchstabens nach einem Passwort fragen
- Beim Anschließen eines Laufwerks einen aussagekräftigen Balloontip anzeigen lassen
- Ausführen eines <u>Autostarts</u>, auch abhängig von o.g. Kriterien
- u.v.a.

Statt Buchstaben können auch leere NTFS-Ordner als Mountpunkte angegeben werden, was aber einige

Einschränkungen und Probleme mit sich bringt, siehe <u>Konfiguration</u>.

Installation

Einfach die ZIP-Datei entpacken, z.B. nach C:\Programme - ein Ordner USBDLM ist enthalten. Im Ergebnis sollten die Dateien dann z.B. in C:\Programme\USBDLM liegen.

Ab Vista/Windows 7 wird unter C:\Programme die USBDLM.INI kommentarlos in den "Virtual Store" umgeleitet falls sie ein eingeschränkter Nutzer bearbeitet, z.B. wenn man die USBDLM.INI im Windows-Explorer doppelklickt. Hier stattdessen immer die _edit_ini.cmd ausführen, die fragt nach den nötigen Admin-Rechten.

Oder USBDLM in einen Ordner wie C:\Tools installieren, hier gibt es keine Umleitung. USBDLM setzt dann die Rechte so wie unter C:\Programme - nur Administratoren <u>und</u> "LocalSystem" haben Schreib- und Ändern-Zugriff. Sonst könnte böswillige Software die USBDLM.EXE ersetzen oder die USBDLM.INI modifizieren, um an höhere Rechte zu gelangen.

USBDLM muss auf einem lokalen Laufwerk installiert sein, es funktioniert nicht, wenn es von einem Netzoder Subst-Laufwerk gestartet wird.

Dann in die USBDLM.INI die gewünschten LW-Buchstaben eintragen. Es ist eine Beispiel-INI enthalten - die USBDLM_sample.INI. Entweder eine neue USBDLM.INI anlegen oder die USBDLM_sample.INI in USBDLM.INI umbenennen und bearbeiten.

Die USBDLM_full.INI zeigt wirklich alle Parameter die USBDLM kennt. Nur für die die statt Doku lieber INI-Dateien lesen...

Zum Registrieren des USBDLM-Dienstes einfach die USBDLM.EXE mit dem Parameter -register starten. Genau das macht die _service_register.cmd

Damit werden keinerlei Dateien kopiert. USBDLM wird nur als Dienst registriert und gestartet.

Ohne Admin-Rechte startet sich die USBDLM.EXE selbst nochmal und fordert dabei die nötigen Admin-Rechten an.

USBDLM trägt sich mit der Startart 'Automatisch' ein - wird also bei jedem Systemstart automatisch vom Service Control Manager gestartet.

Deinstallation:

Der Dienst kann mit dem Parameter -deregister gestoppt und deregistriert werden - das macht die

_service_deregister.cmd

Der Unterstrich am Beginn des Dateinamens dient allein dazu, die Command-Script-Dateien bei alphabetischer Sortierung am Anfang zu haben...

Beides geht auch ohne MessageBox:

USBDLM -silentregister USBDLM -silentderegister

Es gibt dann bei Erfolg Errorlevel 0, sonst 1 zurück.

Der registrierte USBDLM-Dienst kann mit _service_stop.cmd und _service_start.cmd gestoppt und gestartet werden.

Auf der Kommandozeile kann man auch das Windows-Kommando NET bemühen:

net stop USBDLM net start USBDLM

Hinweis: Wenn der USBDLM-Dienst gestartet wird, wird die USBDLM.EXE aus dem Pfad genommen, aus dem der Dienst installiert wurde! Von wo aus der Dienst gestartet wird, spielt keine Rolle, soll ein neues Verzeichnis benutzt werden, dort einmal _service_deregister und dann _service_register ausführen.

Die Laufwerks-Einstellungen in der USBDLM.INI können geändert werden, ohne den Dienst neu zu starten - die INI wird bei jedem Anschließen eines Laufwerks gelesen, wenn sich ihr Datei-Datum/Uhrzeit geändert hat.

Mehr dazu unter Konfiguration.

Hinweis:

Die CMD-Dateien enthalten einfache Kommandozeilen wie "%~dp0usbdlm.exe" -register

%~dp0 ist ein Platzhalter für Laufwerk und Pfad zur CMD-Datei. Das funktioniert so: %0 ist der Platzhalter für die laufende CMD-Datei, z.B. C:\Program Files\USBDLM_service_register.cmd. ~dp ist ein "Modifizierer" und liefert Laufwerk plus Pfad (drive + path), z.B. C:\Program Files\USBDLM\.

Mehr über Batch-Parameter und Modifizierer:

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

History

V5.2.5 (31. August 2016)

- Bugfix: Bei open= wurden bei VBS-Dateien keine Parameter übergeben
- Neu: USB Drive Info zeigt Problem-Codes aus dem Gerätemanager in den Baum-Icons
- Neu: Autorun-Ereignisse OnSafeRemoval und OnUnsafeRemoval

V5.2.4 (22. Juni 2016)

- Bugfix: Seit V5.1.7 funtionierte [OnDeviceArrival] nicht
- Bugfix: SuppressWindowsBalloons=1 funktionierte unter Windows 10 nur auf englischem Windows
- Bugfix: Ballontip-Ausblende-Effekt war unter Vista/Win7,8,10 mit nicht-klassischer Anzeige kaputt
- Bugfix: open= funktionierte nicht für CMD- und BAT-Dateien falls in Anführungszeichen eingeschlossen
- Neu: Mit SuppressWindowsBalloons=0 werden unter Windows 10 USBDLMs Balloontips für Volumes mit Laufwerksbuchstaben um drei Sekunden verzögert, damit der Windows Balloontip Vortritt hat; Die Verzörgerung kann eingestellt werden
- Neu: Suchfunktion in UsbDriveInfo

V5.2.3 (5. Juni 2016)

- Bugfix: Seit V5.2.2 keine Backups der USBDLM.LOG wenn der LogLevel >= 3 ist
- Neu: In der Logdatei kann jeder Zeile das Datum vorangestellt werden (LogShowDate=1)
- Neu: In der Logdatei können Timing, Loglevel und Thread-ID weggelassen werden (LogShowTech=0)

V5.2.2 (28. Mai 2016)

- Bugfix: Unter Vista/Win7/8/10 wurde bei open= der Pfad zur EXE zweimal statt einmal als Parameter übergeben
- Neu: Einträge für **EventLog** Start und Stop des USBDLM-Dienstes
- Neu: Anzahl der Backups für die USBDLM.LOG kann eingestellt werden

V5.2.1 (10. Mai 2016)

• Neu: Einträge für <u>EventLog</u> können einzeln deaktiviert werden

V5.2.0 (8. Mai 2016)

- Bugfix: USBDLM_usr.exe hat (immernoch) manchmal vom Start an permanent CPU-Last verursacht
- Bugfix: Seit V5.0 hat USBDLM hat für neu angeschlossene USB-Laufwerke
 WriteCacheEnableOverride=1 in die Registry geschrieben wenn kein UsbWriteCacheEnable konfiguriert ist

• Neu: <u>EventLog</u> kann alternativ ins LogFile geschrieben werden

V5.1.8 (25. April 2016)

- Bugfix: USBDLM_usr.exe hat manchmal vom Start an permanent CPU-Last verursacht, warscheinlich seit V5.0
- Bugfix: open= hat seit V5.1.3 keine Dokument-Dateien geöffnet
- Bugfix: USB-Speed-Icon wurde an manchen USB-Controllern nicht angezeigt, die nicht mit Microsoft-Treibern laufen
- Bugfix: Maximale USB-Geschwindigkeit wurde bei USB 2.0 High-Speed-Ger\u00e4ten die mit Full-Speed laufen als USB 1.1 Full-Speed angezeigt

V5.1.7 (07. April 2016)

- Bugfix: Ab V5.1.5 hat das Kriterium PortName für USB-Laufwerke erneut nicht funktioniert
- Bugfix: Diverse kleine Probleme behoben
- Neu: OnArrival wird bei Einlegen eines Datenträgers in ein CD/DVD-Laufwerk ausgeführt (bisher nur wenn das ganze Laufwerk angeschlossen wurde oder wenn NoMediaNoLetter für das Laufwerk aktiv war)

V5.1.6 (17. März 2016)

 Bugfix: Seit V5.0 hat USBDLM keine Buchstaben mehr zugeordnet nachdem ein Laufwerk angeschlossen wurde für dessen BusType kein DriveLetters-Abschnitt vorhanden ist

V5.1.5 (11. März 2016)

- Bugfix: In V5.1.4 Crash bei Balloontip mit NTFS-Mountpunkt
- Bugfix: Seit V5.0 haben Einstellungen in der Registry nicht funktioniert

V5.1.4 (09. März 2016)

- Bugfix: In V5.1.3 war die Win32 USBDLM.exe ohne Optimierung für "Code-Größe minimieren" compiliert, neben der ca. 100KB größeren EXE hat das offenbar auch dazu geführt, dass einige Virenscanner glauben, einen Virus zu erkennen
- Bugfix: Ab V5.0 hat das Kriterium PortName für USB-Laufwerke nicht funktioniert
- Rollback: Default für <u>SuppressWindowsBalloons</u> ist wieder 1; 2 (Windows-Balloontips kurzzeitig deaktivieren) führt dazu, dass das Windows Tray-Icon für "Hardware sicher entfernen" nicht mehr reagiert

V5.1.3 (07. März 2016)

• Bugfix: Mehrere Absturzursachen beseitigt

V5.1.2 (03. März 2016)

• Bugfix: Logdatei angelegt trotz WriteLogFile=0

V5.1.1 (29. Februar 2016)

• Bugfix: AutoDismount-TrayIcon wurde auch ohne [AutoDismount]-Abschnitt angezeigt

V5.1.0 (29. Februar 2016)

- Bugfix: In Abschnitten wie [OnArrival] wurden die Zeilen system, elevated und restricted immer als gleich 1 bewertet, falls vorhanden
- Bugfix: In Abschnitten wie [OnArrival] mit nur einer open= Zeile wurden nur Optionen berwertet die nach der open-Zeile stehen
- Neu: Tray-Icon mit Anzeige des <u>Auto-Dismount</u>-Status
- Neu: DismountOnSleep weckt keine schlafenden Laufwerke mehr auf

V5.0.2 (19. Januar 2016)

• Bugfix: In USBDLM.INI auf dem angeschlossenen Laufwerk haben u.U. nur Letter= und Letters= funktioniert, Letter1= usw. aber nicht

V5.0.1 (18. Januar 2016)

- Bugfix: _edit-ini.cmd hat keine USBDLM.INI angelegt, wenn keine Schreibrechte auf das USBDLM-Verzeichnis gegeben waren.
- Bugfix: USB-Geschwindigkeit im Balloontip falsch, wenn die USB Portnummer nicht ermittelt werden konnte, z.B. wenn in der Location nicht wie ab Vista üblich sowas wie Port_#0002.Hub_#0003 steht. Die Portnummer wird jetzt per USB-API ermittelt.
- Bugfix: Abschnitt [NetworkLetters] in V5.0.0 wirkunglos falls kein Abschnitt [DriveLetters] vorhanden

V5.0.0 (17. Januar 2016)

- Neu: _install.cmd und _uninstall.cmd in das umbenannt was sie wirklich tun: _service_register.cmd und _service_deregister.cmd; _start.cmd und stop.cmd wurden umbenannt in _service_start.cmd und _service_stop.cmd damit sie ins Namensschema passen
- Neu: Für INI-Abschnitte und Werte ist keine Nummerierung mehr nötig, es zählt dann allein die Reihenfolge; Mit Nummerierung funktioniert es wie gehabt
- Neu: Ballontips zeigen bei USB-Laufwerken im Icon ob sie mit Full-, High-, SuperSpeed oder SuperSpeedPlus laufen
- Neu: Ballontips zeigen bei USB-Laufwerken im Titel deren USB-Version oder maximale USB-Geschwindigkeit (High-Speed, SuperSpeed usw)
- Neu: <u>Ballontips</u> nach dem Entfernen mit Anzeige ob das Entfernen sicher war oder nicht
- Neu: Konfigurierbare <u>Ballontip-Texte</u>
- Neu: <u>AutoDismount</u>
- Neu: Optionale Kurznotation für <u>Balloontips</u> für Multislot-Cardreader falls maximal einem Slot ein Mountpunkt zugeordnet wurde
- Bugfix: Nach erfolgreicher Passwortabfrage wurde manchmal kein Laufwerksbuchstabe zugeordnet
- Bugfix: Removal Policy wurde in die Registry geschrieben, auch wenn kein

V4.9.1.0 (30. August 2015)

• Bugfix: Starten von cmd- und bat-Dateien funktionierte seit V4.8.8.6 nicht, wenn Leerzeichen im Pfad enthalten sind

V4.9.0.0 (04. August 2015)

- Bugfix: DeviceTypes CardReader und MSCR nicht gesetzt wenn die Disk den DriveType FIXED hat
- Neu: Unterdrücken von Windows-Balloontips funktioniert unter Windows 10

V4.8.9.9 (12. Juni 2015)

- Workaround: Windows 7s Alt-Tab-Taskswitcher mit aktivem Aero kommt durcheinander wenn es
 ein sichtbares TopMost-Fenster mit Größe Null gibt genau so eines nutzte USBDLM um
 Benachrichtigungen über Laufwerksbuchstaben zu empfangen; Jetzt ist es nicht mehr TopMost und
 nicht mehr sichtbar funktioniert genauso gut und Windows 7 ist glücklich
- Workaround: DismountOnSleep=1 hat zu einer zehnsekündigen Verzögerung geführt, wenn USBDLM das "System Restore" Volume dismounten wollte - Ursache unbekannt; USBDLM versucht's bei diesem Volume nicht mehr

V4.8.9.8 (15. Juni 2015)

- Bugfix: Warnungen des PnP-Managers im Windows XP EventLog weil USBDLM Geräte-Benachrichtigungen beim Stoppen des Dienstes zu spät deregistert hat
- Bugfix: Bei abgeschaltetem Windows AutoMount hat USBDLM Volumes nicht online gebracht obwohl ein dafür zuständiger DriveLetters-Abschnitt vorhanden ist
- Bugfix: MD5-Prüfung für open= ging nur wenn MD5-Hash groß geschrieben

V4.8.9.7 (10. Mai 2015)

• Bugfix: Laufwerksbuchstaben von iPods und ignorierten Laufwerken wurden seit V4.8.6 entfernt

V4.8.9.6 (29. März 2015)

Bugfix: UsbWriteProtect hat nicht funktioniert, wenn der Registry-Key
 HKLM\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies nicht vorhanden ist

V4.8.9.5 (09. März 2015)

- Neu: Limits für AutoRun delay und wait auf 24h erhöht; Einheiten können benutzt werden, z.B. 10s statt 10000
- Neu: Bei <u>Buchstaben abhängig von der Zeit</u> können mehrere Zeiträume angegeben werden
- Neu: Buchstaben abhängig von der Zeit auch abhängig vom Wochentag

V4.8.9.4 (15. Februar 2015)

• Bugfix: Aktivierung eines potentiellen BadUSB-Gerätes hat u.U. nicht funktioniert

 Bugfix: Windows hat ab Vista bei aktiver UAC TrueCrypt-Laufwerke manchmal lange nach dem Mounten nochmals gemeldet; hier überspringt USBDLM jetzt Balloontip und AutoRuns

V4.8.9.3 (02. Februar 2015)

• Bugfix: Seit V4.8.9.2 Crash USBDLM beim Ausführen eines Systemevents

V4.8.9.2 (01. Februar 2015)

• Bugfix: Gelegentlich Crash USBDLM bei Ankunft eines TrueCrypt-Volumes

V4.8.9.1 (18. Januar 2015)

- Bugfix: Gelegentlich Crash USBDLM bei Ankunft eines Volumes
- Bugfix: Zugriff auf Floppy-Laufwerk bei Start von USBDLM unter XP entfernt

V4.8.9.0 (28. Dezember 2014)

- Bugfix: Seit V4.8.4 wurden virtuelle Volumes (z.B. Subst- und Netzlaufwerke) ab Vista mit aktiver
 UAC nicht korrekt behandelt
- Bugfix: Laufwerksgröße im Balloontip wird wieder auf ganze Gigabytes gerundet
- Neu: Unterstützung für TrueCrypt-Ersatz VeraCrypt
- Neu: UsbDriveInfo kann NTFS-Mountpunkte löschen und hinzufügen

V4.8.8.8 (10. November 2014)

- Bugfix: Beim Anlegen einer initialen USBDLM.INI wurde NoMediaNoLetter=0 doppelt eingetragen
- Neu: DeleteSviFolder funktioniert auch unter anderen Systemen als XP

V4.8.8.7 (14. Oktober 2014)

- Bugfix: Abschnittsname f
 ür Keep Alive war falsch als DriveKeepAlive dokumentiert, richtig ist DiskKeepAlive
- Bugfix: USBDLM hat manchmal unnötige No-Letter-Einträge in der MountManager-Datenbank angelegt

V4.8.8.6 (12. Oktober 2014)

- Bugfix: Bei AutoRun von CMD und BAT-Dateien (z.B. open="C:\Batch\test.cmd") wurden seit V4.8.8.0 keine Parameter übergeben
- Bugfix: überspringen von AutoRun durch Halten der Shift-Taste hat ab Vista nicht funktioniert;
 "Lösung": Force=1 ist ab Vista Default (Shift-Taste wird ignoriert) und USBDLM_usr.exe wird zum Status der Shift-Taste befragt wenn Force=0 für eine open= Zeile gesetzt ist
- Bugfix: UsbDriveInfo x64 hat diverse Debug-Infos gezeigt die nicht für den Endnutzer gedacht waren
- Bugfix: Balloontiptext für mehrere Partitionen wurde genutzt auch wenn nur eine Partition mit Volume vorhanden ist
- Neu: Keep Alive für Laufwerke

V4.8.8.5 (19. September 2014)

- Bugfix: Bei erneutem Anschließen eines Laufwerks wurden u.U. keine Buchstabe zugeordnet und Autoruns ausgeführt (wenn Windows den DevicePath eines zuvor angeschlossenen Laufwerks recycelt aber einen neuen VolumeName vergibt)
- Bugfix: Mehrere FIXED Laufwerke an einem Controller werden nicht mehr in einem gemeinsamen BalloonTip angezeigt

V4.8.8.4 (17. September 2014)

 Bugfix: Bei erneutem Anschließen eines Laufwerks wurden u.U. Autoruns nicht ausgeführt und Balloontips nicht angezeigt (wenn DevicePath und VolumeName wie zuvor, DeviceInstance-Handle aber abweichend, nicht für möglich gehalten, scheint aber vorzukommen)

V4.8.8.3 (27. August 2014)

• Bugfix: Crash bei sehr langen Volume-Kernelnamen

V4.8.8.2 (21. August 2014)

- Bugfix: Behandlung von potentiellen "<u>BadUSB</u>"-Geräten hat unter Vista+ nur funktioniert, wenn ein Abschnitt [OnDeviceArrival] vorhanden ist
- Bugfix: [NetworkLetters] hat seit V4.6 Laufwerksbuchstaben in [DriveLetters] nicht (wie dokumentiert) "überstimmt"

V4.8.8.1 (19. August 2014)

- Bugfix: Seit V4.8.8.0 wurde die Logdatei immer nach C:\ USBDLM.LOG geschrieben
- Bugfix: UsbDriveInfo hat bei Refresh Änderungen der USBDLM.INI nicht gelesen

V4.8.8.0 (17. August 2014)

- Bugfix: Seit V4.8.7 war das Kriterium USB-Port unzuverlässig
- Bugfix: Seit V4.8.7 wurden neu angeschlossene Laufwerke u.U. nicht korrekt behandelt, wenn zuvor ein Laufwerk mit identischer Volume-Device-ID angeschlossen war
- Bugfix: Seit V4.8.7 wurde beim Anschließen eines Laufwerks dessen Laufwerksbuchstabe kurz entfernt, auch wenn das nicht nötig war
- Bugfix: Seit V4.5.9 wurden OnUserLogon-Abschnitte nur für den ersten gleichzeitig angemeldeten Nutzer ausgeführt
- Bugfix: USBDLM hat beim Registrieren als Windows-Dienst mittels der Windows-Funktion
 CreateService beim Parameter lpBinaryPathName den Pfad zur USBDLM.exe nicht in
 Anführungszeichen eingeschlossen wenn der Pfad Leerzeichen enthält. lpBinaryPathName wird vom
 Service Control Manager aber als Kommandozeile ausgewertet, bei C:\Program
 Files\USBDLM\USBDLM.exe hätte er eine vorhandene C:\Program.exe mit Systemrechten
 ausgeführt. Das MSI hat es richtig gemacht. USBDLM korrigiert den Fehler ohne Neuinstallation.
- Neu: Einstellung SafeCommandLines um mehrdeutige Kommandozeilen zurückzuweisen
- Neu: UsbDriveInfo warnt vor Kommandozeilen die Leerzeichen enthalten, die EXE aber nicht in

Anführungszeichen eingeschlossen, die Kommandozeile somit mehrdeutig ist

 Neu: Rückfrage bei Anschluss einer USB-Tastatur und USB-Netzwerkadapter zum Schutz vor "BadUSB"-Geräten

V4.8.7.0 (29. Juni 2014)

- Bugfix: Seit V4.8.0 Crash USBDLM wenn beim Logout ein virtuelles Volume (Subst-, Netz- oder TrueCrypt-Laufwerk) vorhanden ist und bei einigen anderen Gelegenheiten
- Bugfix: In UsbDriveInfo blieb die Baumansicht u.U. leer
- Neu: UsbDriveInfo behält zum "sicheren Entfernen vorbereitete" Geräte in der Liste
- Neu: UsbDriveInfo erkennt Datenträgerwechsel per Benachrichtigung anstatt im Sekundentakt zu pollen
- Neu: UsbDriveInfo kann ab Windows 7 VHDs ein- und aushängen, unter XP und Vista wenn VHDmount aus Microsoft Virtual Server 2005 in dessen Standardordner installiert ist, die vhdmount.dll also in %ProgramFiles%\Microsoft Virtual Server\Vhdmount liegt

V4.8.6.0 (24. April 2014)

- Bugfix: Mehrere Absturzursachen beseitigt
- Neu: USBDLM-Variable %DateISO%, z.B. 2014-04-15
- Neu: USBDLM versucht schon beim Ankunft der Disk dem Mountmanager Daten für das Volume unterzuschieben - so wird das Problem beseitigt, dass Windows zunächst einen nicht gewünschten Buchstaben zuordnet den USBLDM dann wieder ändert. Ab Windows 8 klappt das in der Regel nicht, das ist beim Zuordnen zu schnell. Mit GPT partitionierten Laufwerken funktioniert es generell nicht.
- Neu: VolumeSize, VolumeSerial und DeviceType TrueCrypt werden anhand der Partitionsinfos statt anhand des Volumes ermittelt -> vermeidet manchmal Verzögerungen
- Neu: Nach dem Anschließen greift USBDLM nicht mehr auf das Volume zu, außer es ist zum Ermitteln von definierten Kriterien unvermeidlich
- Bugfix: Seit V4.7.3 Crash UsbDriveInfo wenn der USBDLM-Dienst auf einem Nicht-NTFS-Laufwerk installiert ist
- Bugfix: UsbDriveInfos Reports waren bei 32KB abgeschnitten
- Neu: Report-Format von UsbDriveInfo überarbeitet
- Neu: UsbDriveInfo Kommandozeilenparameter -rv -rd -rb zum Speichern eines Reports der Ansicht Volumes, Drives oder BusTypes (ein großes R schreibt eine Unicode-Datei, sonst ANSI)
- Neu: UsbDriveInfo kann Laufwerksbuchstaben ändern falls mit Adminrechten gestartet

V4.8.5.0 (29. Januar 2014)

- Bugfix: Seit V4.8.2 USBDLM hat ein "Friendly Name" nicht als DeviceID-Kriterium funktioniert
- Bugfix: UsbDriveInfo hat in der BusTypes-Ansicht keine Netzlaufwerke angezeigt

V4.8.4.0 (28. Januar 2014)

- Bugfix: Seit V4.8.0 u.U. Crash USBDLM bei NoMediaNoLetter>0
- Neu: UsbDriveInfo mit BusTypes-Ansicht

V4.8.3.0 (23. Januar 2014)

- Bugfix: Seit V4.8.0 Crash USBDLM bei Anschließen eine Laufwerks das nicht innerhalb von 2 Sekunden antwortet
- Bugfix: Absturz UsbDriveInfo wenn keine USBDLM.INI
- Bugfix: Kommandoschnittstelle lieferte bei Erfolg u.U. einen Fehler zurück

V4.8.2.0 (12. Januar 2014)

- Bugfix: Seit V4.8 Absturz USBDLM und UsbDriveInfo nach vielmaligem Anschließen und Trennen eines Laufwerks
- Bugfix: CPU-Last nach Nutzung der Kommandoschnittstelle
- Bugfix: OnServiceStart wurde nie ausgeführt
- Bugfix: Seit V4.8 zeigte UsbDriveInfo u.U. nicht die korrekten INI-Abschnitte

V4.8.1.0 (27. Dezember 2013)

• Bugfix: Balloontip für U3-Sticks hat in V4.8.0.0 meist nur ein Laufwerk angezeigt

V4.8.0.0 (25. Dezember 2013)

- Bugfix: Überprüfung der Laufwerksbuchstaben beim Dienst-Start hat für Nicht-USB-Laufwerke nicht funktioniert wenn nur eine DeviceID konfiguriert ist und der BusType daraus nicht offensichtlich hervorgeht, z.B. ein Laufwerk mit BusType SATA und DeviceID SCSI\...
- Bugfix: MD5-Prüfung hat zuletzt in V4.2.1 funktioniert
- Bugfix: Der von Windows zugeordnete Laufwerksbuchstabe von in [IgnoreDevices] konfigurierten Laufwerken wurde entfernt
- Bugfix: Das fest eingebaute Ignorieren des Apple iPod hat nicht funktioniert
- Neu: Events für Disks: [OnDiskArrival], [OnDiskRemovalRequest], [OnDiskRemoval]
- Neu: <u>Kommando-Schnittstelle</u> für eingeschränkte Nutzer zum Flushen des Datei-Schreibcaches, Dismounten, Hardware sicher entfernen, u.a.
- Neu: UsbDriveInfo zeigt Copy&Paste-taugliche Kriterien (Options -> Show USBDLM Criteria)
- Neu: UsbDriveInfo Infos umgestaltet und erweitert
- Neu: UsbDriveInfo ermittelt den Container von TrueCrypt-Volumes und kann diese "auswerfen"
- Entfernt: Unterstützung von Windows 2000

V4.7.3.2 (09. Juni 2013)

• Bugfix (nochmal): Crash auf manchen Rechnern

V4.7.3.1 (22. Mai 2013)

• Bugfix: Crash auf manchen Rechnern

V4.7.3.0 (20. Mai 2013)

- Bugfix: Crash wenn Nutzer Mitglied eine großen Anzahl von Nutzergruppen ist (~ >100)
- Neu: ListUsbDrives Kommandozeilen-Programm durch UsbDriveInfo GUI-Programm ersetzt;
 ListUsbDrives ist weiterhin einzeln erhältlich
- Neu: Beim Kriterium UserGroup kann eine Gruppen-SID statt eines Gruppennamens angegeben werden
- Neu: <u>Gruppenzugehörigkeit</u> wird über das Access-Token des Nutzers ermittelt; Das funktioniert auch bei Login über "cached credentials"; NetUserGroups=1 ist nicht mehr nötig
- Neu: <u>Kommando-Schnittstelle</u> zum Flushen des Schreibecaches, Dismounten, Auswerfen und reaktivierbares "Hardware sicher entfernen"
- Neu: <u>DeviceType</u> "WindowsRecovery" für Windows Recovery-Partitionen
- Neu: USBDLM stellt sicher, dass USBDLM.EXE und INI nur von Administratoren geschrieben werden können
- Neu: _edit-ini.cmd um USBDLM.INI mit Admin-Rechten zu öffnen

V4.7.2.0 (10. April 2013)

- Neu: Ausgaben von versteckt oder im Kontext "System" gestarteten Programmen werden in die Logdatei umgeleitet
- Bugfix: Crash bei mehr als 160 Laufwerken einer Klasse
- Bugfix: USB-Portname wurde nicht ermittelt wenn das Laufwerk Teil eines USB-Verbundgerätes (USB composite device) ist
- Bugfix: u.U. Crash beim Start des USBDLM-Dienstes
- Bugfix: Crash ListUsbDrives wenn eine Partition eine Ausrichtung >= 4 GB hat
- Bugfix: Bei Anschließen einer USB-Floppy wurde bei V4.7.1 kurzzeitig ein falscher Balloontip angezeigt
- Bugfix: Crash beim Entfernen eines Laufwerks wenn beim Anschließen der VolumeName nicht ermittelt werden konnte
- Bugfix: Warnung bei USBDLM.INI im "Virtual Store" von Vista/Win7/Win8 hat nicht funktioniert
- Bugfix: ListUsbDrives hat ohne Admin-Rechte den USBDLM-Dienst-Status nicht angezeigt

V4.7.1.0 (09. September 2012)

- Bugfix: Crash wenn ein Laufwerk auf den ersten freien Buchstaben umgemountet werden soll und der LogLevel >= 2 ist
- Bugfix: Nach Standby und Ruhezustand wurden Laufwerke ohne eingelegten Datenträger aus USBDLMs interner Liste entfernt. Als Folge funktionierten für diese Laufwerke kein OnRemovalRequest und OnRemoval mehr.
- Bugfix: Unter Windows 2000 funktionierten OnArrival und BalloonTips bei machen Geräten nicht
- Bugfix: Keine <u>Balloontips</u> für TrueCrypt-Volumes mit ShowTrueCrypt=1 und ShowVirtual=0
- Bugfix: AutoRuns beim Systemstart u.U. nicht ausgeführt wenn AutoRunOnLogon=1 und AutoRunOnStartup=0
- Neu: Buchstaben abhängig von der <u>Disk-Signatur</u>

- Neu: <u>StaticLetters</u> Laufwerksbuchstaben die USBDLM unverändert lässt. So kann man trotz USBDLM noch bestimmte Laufwerksbuchstaben mit der Windows Datenträgerverwaltung zuordnen ohne dass USBDLM das rückgängig macht
- Neu: Gerätetyp OtfeVolume für virtuelle Laufwerke erstellt von <u>FreeOTFE</u>
- Neu: In <u>OnDeviceArrival</u>-Abschnitten sind alle Kriterien verfügbar, die nicht im Zusammenhang mit einem neuen Laufwerk stehen
- Neu: Das Löschen des Ordners System Volume Information funktioniert auf NTFS-Laufwerken
- Neu: Einige Optimierungen

V4.7.0.0 (15. Januar 2012)

- Bugfix: Kriterium MaxTimeSinceBoot hat nicht funktioniert
- Neu: Buchstaben abh. vom USB-Port funktioniert auch mit Nicht-USB-Laufwerken
- Neu: <u>Buchstaben abh. vom Betriebssystem</u>
- Neu: <u>Textdateien mit Geräte-IDs</u> werden im Speicher gehalten statt jedes Mal neu eingelesen
- Neu: ListUsbDrives zeigt (mit -a gestartet) Filtertreiber für Laufwerke und die zugehörigen Geräteklassen
- Neu: ListUsbDrives zeigt die Ausrichtung von Partitionen und FAT-Clustern
- Neu: Warn-Balloon-Tip wenn unter XP eine Festplatte mit GUID-Partition-Table (GPT) angeschlossen wird

V4.6.9.2 (08. Oktober 2011)

- Bugfix: Seit V4.6 hat autorun.inf open meist nicht funktioniert
- Neu: Timeout-Zeit bei Kriterien FileExists, DirExists usw. konfigurierbar
- Neu: Registry-Wert für USB WriteProtect kann gesetzt werden

V4.6.9.1 (11. September 2011)

- Bugfix: Ab V4.6 funktionierte die externe Textdatei mit Geräte-IDs nur als Unicode
- Neu: Zeit als Kriterium
- Neu: Win7 BusTypes Virtual and FileBackedVirtual

V4.6.8.3 (10. Juli 2011)

- Bugfix: Ab V4.6 Crash beim Installieren wenn WriteLogFile=1 und die Log-Datei nicht angelegt werden kann, z.B. weil die nötigen Rechte fehlen oder der Pfad ungültig ist
- Bugfix: Crash wenn FileExists-Kriterium bei System- und Geräteereignissen ausgewertet wurde

V4.6.8.2 (19. Juni 2011)

• Bugfix: FireWire-Laufwerke hatten ab V4.6 verstümmelten FriendlyName, u.a. im BalloonTip

V4.6.8.1 (08. Juni 2011)

• Bugfix: Einige Funktionen haben in V4.6.8.0 nur eine begrenzte Zeit funktioniert

V4.6.8 (05. Juni 2011)

• Bugfix: Kriterium UserGroup hat auf manchen Vista/Win7-Installationen ab V4.6 nicht funktioniert

V4.6.7 (30. Mai 2011)

Bugfix: Passwort-Abfrage hat unter Vista/Win7 ab V4.6 nicht funktioniert

V4.6.6 (27. Mai 2011)

• Bugfix: Abschnitt NetworkLetters/ExcludedLetters war ab V4.6 wirkungslos

V4.6.5 (15. Mai 2011)

- Bugfix: Ab V4.6 möglicher Absturz wenn mehrere Netz- oder Subst-Laufwerke gleichzeitig angelegt werden
- Neu: <u>Setting Virtual Volumes zum Deaktivieren der Unterstützung für virtuelle Laufwerke</u> (Netz, Subst, TrueCrypt u.a.)

V4.6.4 (26. April 2011)

Bugfix: Einige kleine Bugfixes

V4.6.3 (21. März 2011)

- Bugfix: USB-Laufwerke auf "Für Leistung optimieren" und Schreibcache erzwingen hat nur funktioniert, wenn für das Laufwerk der Registry-Schlüssel ClassPnP bereits vorhanden war, was nur der Fall ist, wenn man die Richtlinie für das sichere Entfernen schon mal umgestellt hatte
- Neu: Neue Variable %PureVolumeName% liefert sowas wie Volume{06588842-39a4-11e0-8027-806d6172696f}, was nützlich für Tools wie DD ist, die Volume-Namen im Format \\.\Volume{06588842-39a4-11e0-8027-806d6172696f} sehen wollen

V4.6.2 (15. März 2011)

• Bugfix: Ab V4.6 wurden Registry-Einstellungen nicht korrekt gelesen

V4.6.1 (23. Februar 2011)

 Bugfix: Ab V4.6 keine Balloontips unter Windows Server 2008 sowie unter Windows Vista und Windows 7 bei deaktivierter UAC

V4.6.0 (20. Februar 2011)

- Neu: Umstellung auf Unicode: USBDLM.INI, BalloonTip, Dateinamen usw. haben keine Probleme mehr mit Codepages
- Neu: OnUserActivated und OnUserDeactivated <u>System-Ereignisse</u>
- Neu: Explorer Icon und Label können alternativ in DriveLetters-Abschnitten definiert werden
- Neu: Ausführbare Dateien werden zuerst im Windows-Suchpfad (Umgebungsvariable PATH) gesucht, dann im Arbeitsverzeichnis
- Neu: OnArrival und OnRemoval-Events für Netz- und Subst-Laufwerke
- Neu: Das Kriterium FileExists sieht nutzerspezifische Laufwerke (Netz-, Subst und TrueCrypt-Laufwerke)

- Neu: Laufwerksbuchstabe kann aus der <u>Datenträgerbezeichung</u> abgeleitet werden
- Neu: ListUsbDrives zeigt mit Parameter -v gestartet alle USBDLM-<u>Variablen</u>
- Neu: USBDLM kann <u>Schreibcache für USB-Laufwerke</u> aktivieren
- Neu: Ein x64-MSI ist verfügbar
- Neu: <u>USBDLM.WXS</u>-Datei zur Erzeugung einer MSI-Datei mit dem <u>Microsoft WIX</u> Toolset

V4.5.9 (20. Februar 2011)

 Bugfix: Ab V4.5.1 hörte USBDLM unter folgenden Bedingungen auf zu arbeiten: USB-Laufwerk wird angeschlossen, dass als Kartenleser erkannt wird, ein [Explorer] Abschnitt ist vorhanden, für das Laufwerk gibt es keinen zuständigen Abschnitt [OnArrival], [OnRemovalRequest] und [OnRemoval]

V4.5.8 (17. November 2010)

- Bugfix: Ab V4.5 haben USB-Floppy-Laufwerke u.U. den ersten freien Buchstaben größer C bekommen, statt A oder B zu behalten
- Bugfix: Ab V4.5.3 hat USBDLM die Windows Timerauflösung von 10 oder 15 auf 1 ms erhöht die Folge könnte eine minimal höherer Leistungsaufnahme der CPU sein

V4.5.7 (6. November 2010)

- Bugfix: Ab V4.5.3 haben Einstellungen in der Registry nicht funktioniert
- Bugfix: Ab V4.5.3 haben Einstellungen in der INI nicht funktioniert, wenn der letzte Abschnitt einen unbekannten Namen hat

V4.5.6 (14. Oktober 2010)

- Bugfix: Bei fünf oder mehr Laufwerken fehlte beim V4.5.5 im Balloontip das Wort "Laufwerke"
- Neu: Innerhalb einer Sekunde mehrfach gemeldete DeviceArrivals werden ignoriert

V4.5.5 (12. Oktober 2010)

- Bugfix: OnDeviceArrival hat in USBDLM V4.5.4 nicht funktioniert
- Neu: Etwas Grammatik für Balloontips in slawischen Sprachen

V4.5.4 (10. September 2010)

• Bugfix: In USBDLM V4.5.3 funktioniert OnBallonClick nach Standby und Ruhezustand nicht mehr

V4.5.3 (5. September 2010)

- Bugfix: USBDLM V4.5.2 verhinderte das Zurückschreiben von Roaming User Profiles auf den Server
- Bugfix: Text_mounted_to_plr für die Balloontips funktioniert
- Neu: Dock- und Undock <u>System-Ereignisse</u>
- Neu: Einlesen komplexer INI-Dateien beschleunigt
- Neu: Einige Optimierungen

V4.5.2 (1. August 2010)

- Bugfix: Verstümmelte Balloontips für Geräte mit Anführungszeichen im Namen
- Bugfix: OnArrival-Abschnitte u.U. beim Systemstart doppelt ausgeführt
- Bugfix: Verzögerung beim USBDLM-Dienststart beseitigt
- Bugfix: USBDLM Timeout-Fehler im Ereignisprotokoll auf manchen langsam startenden Systemen
- Bugfix: zunächst nicht antwortende Laufwerke mit Wechseldatenträger wurden nicht erneut geprüft,
 OnArrival-Abschnitt nicht ausgeführt
- Bugfix: BitLocker-Laufwerke u.U. nicht als solche erkannt
- Bugfix: Einige Bugfixes für Windows 2000
- Neu: Buchstaben abhängig vom Computernamen
- Neu: Buchstaben abhängig von einem laufenden oder nicht laufenden Prozess
- Neu: ListUsbDrives zeigt USB 3.0 SuperSpeed

V4.5.1 (13. Juni 2010)

- Bugfix: Nach dem Wiederaufwachen aus Standby oder Ruhezustand wurden nicht antwortenden Kartenlesern Buchstaben zugeordnet
- Bugfix: Wurde in einem [OnRemovalRequest] ein TrueCrypt-Volume abgemeldet, hat USBDLM die Nachricht dazu nicht verarbeitet und beim n\u00e4chsten Mounten deshalb kein [OnArrival] und die autorun.inf verarbeitet und keinen Balloontip angezeigt
- Bugfix: Der Passwort-Dialog wurde auf manchen Systemen nicht angezeigt
- Bugfix: ListUsbDrives ist mit der aktuellen usb.id-Datei abgestürzt
- Neu: Passwort kann mit anonymer Anmeldung über's Netzwerk empfangen werden

V4.5.0 (9. Mai 2010)

- Neu: Autorun bei System-Ereignissen abhängig vom Nutzer
- Neu: Vista <u>Scan and Fix</u> kann unterdrückt werden
- Neu: Passwortabfrage für Laufwerke
- Neu: AutoRun funktioniert für Remote-Nutzer
- Neu: AutoRun für beliebige Geräte

V4.4.2 (13. Dezember 2009)

- Bugfix: Variable %UsbSerial% war ab V4.4.0 immer leer
- Bugfix: Tausch von Laufwerksbuchstaben wurde nicht durchgeführt wenn der USBDLM-Dienst nach dem Login startet
- Bugfix: FileExists funktionierte bei Einsatz von Wildcards nur wenn exakt eine Datei gefunden wurde
- Neu: Tausch von Laufwerksbuchstaben beim Dienst-Start kann Ringtausch durchführen

V4.4.1 (11. November 2009)

• Bugfix: Ausblenden von Laufwerken in <u>"Hardware sicher entfernen"</u> hat u.U. unter Vista nicht

- funktioniert, unter Windows 7 funktioniert es für Wechseldatenträger nach wie vor nicht (und wird's wohl auch nie)
- Bugfix: ListUsbDrives hat immer einer falsche Warnung wegen einer USBDLM.INI im Vista/Win7
 Virtual-Store gezeigt
- Bugfix: Verzögerung beim Dienst-Start auf manchen Systemen
- Bugfix: Schließen-Button auf dem Balloontip unter XP u.U. unvollständig gezeichnet

V4.4.0 (1. November 2009)

- Bugfix: Deaktivieren der Systemwiederherstellung für neue Laufwerke hat nicht funktioniert
- Bugfix: <u>BusTypes</u> in Listennotation (z.B. BusType=USB,SCSI) haben u.U. nicht funktioniert
- Bugfix: OnArrival hat für Laufwerke ohne eingelegten Datenträger auch mit MinDiskSize=- nicht funktioniert
- Bugfix: ListUsbDrives zeigt jetzt alle für ein Laufwerk wirksamen Abschnitte
- Neu: x64-Version verfügbar
- Neu: Start des USBDLM-Dienstes bescheunigt
- Neu: Buchstaben abhängig vom <u>USB-Port</u> arbeitet ohne nennenswerte Verzögerungen
- Neu: Label und Icon aus <u>autorun.inf</u> können genutzt werden
- Neu: In <u>autorun.inf</u> Dateien wird auf x64-Systemen zuerst der Abschnitt [autorun.amd64] gelesen
- Neu: <u>autorun.inf</u> kann auch dann gelesen werden, wenn sie per <u>IniFileMapping</u> in die Registry umgeleitet wurde
- Neu: Das Kriterium VolumeSerial kann mit Wildcards benutzt werden
- Neu: "BitLocker To Go" verschlüsselte Laufwerke werden als solche erkannt
- Neu: Zusätzlicher Suchpfad für ausführbare Dateien
- Neu: Alternative Notation des <u>USB-Portnamens</u>

V4.3.2 (7. Juli 2009)

- Bugfix: Wenn in einem Abschnitt nur NTFS-Mountpunke konfiguriert waren, wurde ein vorhandener Laufwerksbuchstabe nicht entfernt
- Bugfix: Textfarbe im Balloon-Tip war immer schwarz, auch wenn in Windows anders konfiguriert
- Bugfix: ListUsbDrives zeigt für Festplatten und Wechseldatenträger den richtigen "Disk DosDevName" (bisher war's der Physical Device Object (PDO) Name)
- Neu: Beim Start kann USBDLM Laufwerksbuchstaben u.U. tauschen, bisher musste der Zielbuchstabe frei sein
- Neu: Abschnittsnummern von 1 bis 9 können als 01 .. 09 notiert werden, z.B. [DriveLetters01]
- Neu: Einträge Letter, Letters, Letter1 bis Letter9 alle verstehen alle Formate (Buchstabe, Buchstaben-Liste, Mountpunkt, INI-Datei)

V4.3.1 (1. Juni 2009)

- Bugfix: Verzögerung bei Standby und Ruhezustand beseitigt
- Bugfix: Langsam antwortende Laufwerke werden toleranter behandelt
- Bugfix: Firewire-Laufwerke haben wieder den kurzen, "handgemachten" Namen im Balloontip

V4.3.0 (30. April 2009)

- Bugfix: Volume-Label nach Medienwechsel wurde unter Vista manchmal nicht erkannt, siehe auch http://support.microsoft.com/kb/943632
- Neu: ListUsbDrives zeigt, welche Abschnitte in der USBDLM.INI für ein Laufwerk zum Zuge kommen
- Neu: Nur wirklich benötigte Laufwerks-Infos werden ermittelt
- Neu: Mehrere <u>NTFS-Ordner</u> können als Mountpunkt zugeordnet werden, auch zusätzlich zu einem Laufwerksbuchstaben
- Neu: Bei [OnArrival], [OnRemoval] usw. können bis zu 10 Kommandos ausgeführt werden (open1 bis open9 und das bisherige open)
- Neu: [OnArrival], AutoRun.inf und BalloonTips für TrueCrypt-Volumes
- Neu: Wenn eine <u>DeviceID</u> konfiguriert ist, kann der BusType für Nicht-USB-Geräte weggelassen werden, ebenso der DriveType z.B. für CDROM-Laufwerke
- Neu: [OnArrival] kann unter Vista Programme mit vollen Rechten ("elevated") starten
- Neu: Kriterium <u>UserIsAdmin</u> bezieht sich unter Vista auf den "elevateten" Nutzer, ist also für einen Admin mit UAC gleich 1
- Neu: Kriterium für <u>Abwesenheit einer Datei</u> (FileNotExists)
- Neu: [OnRemovalFailed] wird asynchron ausgeführt, um Event-Pingpong bei Nutzung von RemoveDrive zu vermeiden
- Neu: AutoRun bei System-Ereignissen
- Neu: Verhalten beim erstmaligen Anschließen von Laufwerken unter Windows 2000 verbessert
- Neu: NoMediaNoLetter zuverlässiger unter Windows 2000
- Neu: Bei OnArrival, OnRemoval usw. gilt das Standard-Kriterium MinDiskSize=1, d.h. es muss ein Medium eingelegt sein
- Neu: Laufwerksbuchstaben können beim <u>Aufwachen</u> aus Standby und Ruhezustand überprüft werden
- Neu: Laufwerksbuchstaben abhängig von der <u>Volume-Seriennummer</u>
- Neu: Laufwerksbezeichnung für USB-Laufwerke im Menü "Hardware sicher entfernen"
- Neu: Ballons unter Vista im fast korrekten Design
- Neu: Balloontips mit Schatten und Ausblenden (jeweils falls in Windows aktiviert)
- Neu: Bei Nutzung von <u>USBDLM-Variablen</u> als Umgebungs-Variablen kann (und sollte) eine Liste der benötigten Variablen konfiguriert werden
- Neu: Abschnitt [ExcludedLetters] umbenannt in [NetworkLetters]

- Bugfix: Verstümmelter BalloonTip wenn Anführungszeichen im Gerätenamen
- Bugfix: Gelegentlich falsche Datenträger-Größe im BalloonTip
- Bugfix: Das Ausführen der <u>autorun.inf</u> Open-Zeile wurde in V4.2.4 (wieder) standardmäßig ohne Admin-Rechte gemacht (AutoRunInfRestricted war standardmäßig aktiviert)

V4.2.4 (3. Dezember 2008)

- Bugfix: Das Ausführen von [OnArrival] schlug fehl wenn der Pfad Leerzeichen enthielt und Kommandozeilenparameter benutzt wurden
- Das Ausführen von [OnArrival] schlug fehl wenn das Laufwerk keinen Mountpunkt hat und kein Arbeitsverzeichnis (WorkDir) angegeben war
- Bugfix: Unter Windows 2000 wurden die <u>Port-Namen</u> von USB1-Geräten an USB2-Ports nicht ermittelt

V4.2.3 (14. September 2008)

- Bugfix: Das Ausführen von [OnArrival] konnte seit V4.0 trotz Einstellung Force=1 durch Halten der Shift-Taste übergangen werden
- Bugfix: Das Ausführen der <u>autorun.inf</u> Open-Zeile wurde seit V4.2 standardmäßig ohne Admin-Rechte gemacht (AutoRunInfRestricted war standardmäßig aktiviert)
- Bugfix: [NoMediaNoLetterInclude] hat meist nicht funktioniert

V4.2.2 (17. August 2008)

- Bugfix: [OnArrival], [OnRemoval] funktioniert wieder ohne Angabe eines Arbeitsverzeichnisses wenn das Laufwerk keinen Mountpunkt hat
- Bugfix: [OnRemoval] funktioniert wieder ohne Angabe eines Arbeitsverzeichnisses
- Bugfix: [OnRemoval] wird beim Entfernen einer Karte aus einem Kartenleser ausgelöst
- Bugfix: Neue <u>BusTypes</u> SD und MMC (Laptop-Kartenleser) funktionieren
- Bugfix: Virtuelle Laufwerke der Vista "Volumen-Schattenkopie" bekommen keinen Laufwerksbuchstaben mehr
- Bugfix: Verzögerung beim Herunterfahren von Windows beseitigt

V4.2.1 (19. April 2008)

• Bugfix: Bei <u>autorun.inf</u>, [OnArrival] <u>usw.</u> ist das Stammverzeichnis wieder das aktuelle Verzeichnis

V4.2.0 (12. April 2008)

- Bugfix: Probleme mit NTFS-Mountpunkten länger als 51 Zeichen behoben
- Bugfix: AutoRun-Programme schafften es unter Vista nicht in den Vordergrund
- Neu/Bugfix: Auch nicht verbundene Netz-Laufwerke werden berücksichtigt
- Neu: NoMediaNoLetter funktioniert unter Windows 2000
- Neu: Fast alle Einstellungen werden ohne Neustart des USBDLM-Dienstes wirksam
- Neu: Wildcards bei vielen Konfigurations-Parametern möglich
- Neu: Bestimmte Laufwerke können komplett ignoriert werden

- Neu: Laufwerke und sonstige Geräte können aus "Hardware sicher entfernen" entfernt werden
- Neu: Richtlinie <u>"Für Geschwindigkeit optimieren"</u> kann für USB-Laufwerke gesetzt werden
- Neu: Beim <u>AutoRun</u> können USBDLM-Variablen dem ausgeführten Programmen als Umgebungsvariablen zur Verfügung gestellt werden
- Neu: Beim <u>AutoRun</u> kann das Programm ohne Admin-Rechte ausgeführt werden (falls der Nutzer Admin ist), oder auch im Kontext "LocalSystem", falls die Rechte des angemeldeten Nutzers nicht ausreichen
- Neu: Ungenutzte NTFS-Mountpunkte können beim Start des Dienstes entsorgt werden

V4.1.0 (19. Oktober 2007)

- Bugfix: Die letzten zwei Zeilen der USBDLM.INI wurden ignoriert (zwei oder mehr leere Zeilen am Ende haben geholfen...)
- Bugfix: OnRemoval funktioniert auch beim Entnehmen einer Karte aus einem Kartenleser
- Bugfix: Auch mit NoMediaNoLetter=0 hat sich USBDLM für einige Benachrichtigungen registriert, unter Vista64 wurde dadurch das "Vorbereiten zum sicheren Entfernen" verhindert
- Bugfix: USBDLM-Dienst unter Vista nicht mehr als "interaktiv" gekennzeichnet -> keine Fehlermeldungen mehr im Syslog
- Workaround: NoMediaNoLetter funktioniert unter Vista64
- Neu: <u>NoMediaNoLetter</u> kann feiner abgestuft werden
- Neu: <u>NoMediaNoLetter</u> kann für einzelne Geräte erzwungen werden per Abschnitt [NoMediaNoLetterInclude] (analog zu [NoMediaNoLetterExclude])
- Neu: Laufwerksbuchstaben-Liste in <u>Kurz-Notation</u>, z.B. Letters=R,X,Y,Z
- Neu: User=- als Kriterium für "kein Nutzer angemeldet"

V4.0.2 (26. August 2007)

- Bugfix: Die in eine Datei umgeleiteten Ausgaben von ListUsbDrives hatten ein falsches Format für den Zeilenvorschub, das der Windows Editor nicht vernünftig anzeigen kann
- Bugfix: AutoRun-Ereignisse für Laufwerke ohne Laufwerksbuchstaben funktionierten nur mit dem zusätzlichen Kriterium Letter=-
- Bugfix: Unter Windows 2000 haben Partitionsnummern und Balloontips bei Festplatten mit mehreren Partitionen nicht funktioniert

V4.0.1 (20. Juli 2007)

- Bugfix: "ExcludedLetters" hat meist nicht funktioniert
- Bugfix: ListUsbDrives konnte ohne Admin-Rechte nicht alle Informationen anzeigen

V4.0.0 (15. Juli 2007)

- Neu: Neues Format für die USBDLM.INI, damit keine festen Prioritäten, mehr Möglichkeiten
- Neu: LW-Buchstaben abhängig von aktiven Nutzer, dessen Gruppe oder dessen Admin-Status
- Neu: AutoRun beim Einlegen eines Mediums (auch CD/DVD)

- Neu: <u>AutoRun-Ereignisse abhängig von allen Kriterien</u> anhand derer USBDLM Laufwerksbuchstaben vergibt
- Neu: AutoRun beim "Vorbereiten zum sicheren Entfernen" und nach dem Entfernen eines Laufwerks
- Neu: Balloontip-Texte in verschiedenen Sprachen integriert
- Neu: AutoRun beim Klick auf den BalloonTip
- Neu: Laufwerks-Ereignisse ins Windows System-Protokoll

V3.4.2 (9.5.2007)

- Bugfix: mehrere (falsche) Buchstaben im Balloon-Tip angezeigt, wenn Hardwareerkennung unvollständig
- Bugfix: Erkennung von U3-Laufwerken verbessert

V3.4.1 (1.5.2007)

- Bugfix: V3.4.0 basiert auf altem Code alle f
 ür V3.4.0 versprochenen Funktionen gehen jetzt wirklich...
- Bugfix: open= in autorun.inf geht jetzt auch ohne Pfadangabe wenn die auszuführende Datei auf dem angeschlossenen Laufwerk liegt

V3.4.0 (18.4.2007)

- Bugfix: Absturz bei mehr als 10 Laufwerken am selben Controller, z.B. bei virtuellen CD-ROMs
- Bugfix: Windows Explorer hat in der Baumansicht Laufwerk u.U. ohne Namen angezeigt
- Neu: LW-Buchstaben abhängig von der LW-Größe auch abh. von der Größe des Datenträgers
- Neu: LW-Buchstaben abhängig vom Laufwerks-Namen ("Volume Label")
- Neu: LW-Buchstaben abhängig von der Geräte-ID kann IDs aus externer Datei laden

V3.3.1 (20.3.2007)

- Bugfix: Möglicher Absturz auf Rechnern mit HyperThreading oder mehreren CPU-Kernen
- Bugfix: Errorlevel bei der Installation mit -silentinstall war immer noch 'falschrum' jetzt 0 bei Erfolg
- Neu/Fix: das Entfernen von Laufwerksbuchstaben für NoMediaNoLetter wir leicht verzögert gemacht, um vereinzelte Probleme mit dem Windows Explorer zu vermeiden
- Neu/Fix: Beim Zuordnen von Laufwerksbuchstaben für NoMediaNoLetter wird das verzögert an den Windows Explorer gemeldet, da er sonst im Verzeichnisbaum den Laufwerksnamen nicht anzeigen will
- Neu/Fix: Bei Deinstallation und NoMediaNoLetter=1 werden Laufwerksbuchstaben von Cardreadern wiederhergestellt

V3.3.0 (08.02.2007)

• Bugfix: Errorlevel bei der Installation waren 'falschrum' - jetzt 0 bei Erfolg

- Neu: kein Problem mit AutoRun mehr unter Vista und Windows 2000
- Neu: Installation, Deinstallation, Start und Stop fordern bei Bedarf Admin-Rechte an
- Neu: <u>AutoRun</u> kann über einen Schlüssel abgesichert werden
- Neu: Kein LW-Buchstabe für Laufwerke die keinen Buchstaben haben und TrueCrypt-Container sind
- Neu: viel Feinschliff :-)
- Neu: statt des guten alten HLP-Format die Hilfe jetzt als CHM, weil Vista HLP standardmäßig nicht unterstützt

V3.2.0 (11.12.2006)

- Bugfix: DeleteMountpointOnRemoval funktionierte nicht seit V3.0
- Neu: 'kein Buchstabe' kann konfiguriert werden, z.B. um bestimmte Laufwerke loszuwerden oder um zu verhindern, dass Nutzer darauf zugreifen
- Neu: Laufwerksbuchstaben können abhängig von der Geräte-ID des Laufwerks konfiguriert werden
- Neu: BallonTips können für bestimmte Laufwerke deaktiviert werden

Hinweis: Diese Version stand bis 6.1.2006 mit der falschen Versionsnummer 3.1.6.1beta zum Download.

V3.1.0 (6.11.2006)

- Bugfix: Balloontips funktionierten nicht für den zweiten angemeldeten Nutzer bei 'schneller Benutzerumschaltung' unter XP und generell unter Vista
- Bugfix: Die Autorun-Funktion hat Programm im Kontext "LocalSystem" statt im Kontext des aktiven Nutzers gestartet
- Neu: Laufwerksbuchstaben für USB-Floppys

V3.0.3 (22.10.2006)

- Bugfix: Mit NoMediaNoLetter=1 ist USBDLM V3.0.2 beim Start abgestürzt...
- Bugfix: Erkennen von Konflikten mit Netz-Laufwerken beim Start hat u.U. nicht funktioniert
- Neu: Multislot-CardReader mit nur zwei Slots werden als solche erkannt, wenn sie wenn sie zwei Slots haben und 'reader' im Geräte-Namen haben
- Neu: IDE- und PCI-CardReader sollten erkannt werden

V3.0.2 (16.10.2006)

 Bugfix: Mit NoMediaNoLetter=1 verloren USB-Floppys ihren Laufwerksbuchstaben, wenn sie ohne eingelegtes Medium angeschlossen wurden

V3.0.1 (7.10.2006)

 Bei AutoRun und RunOnArrival wird das neue Laufwerk zum aktuellen Verzeichnis gemacht, so dass Programme oder Batch-Dateien dort gefunden werden

V3.0.0 (3.10.2006)

- Anwenden der <u>Einstellungen</u> beim Start des Dienstes
- Über Abschnitt [ExcludedLetters] können Buchstaben für die Nutzung durch USB-LWs ausgeschlossen werden
- Lesen von weiteren [ExcludedLetters] aus einer externen USBDLM.INI, z.B. von einen Server
- Behandlung von <u>U3-Autorun-Laufwerken</u>
- Ausführen der open= Zeile in der autorun.inf auf Wechseldatenträgern
- Globales AutoRun beim Anschließen eines beliebigen USB- oder Firewire-Laufwerks
- Balloon-Tip mit Informationen über die zugeordneten Mountpunkte

V2.4.4 (7.9.2006)

• Anpassung: Ein USB-Gerät das zwei Laufwerke enthält, wird nicht mehr als Multislot-Cardreader eingestuft (erst ab drei Laufwerken)

V2.4.3 (28.8.2006)

- Bugfix: Unter Windows 2000 gab's eine Fehlermeldung 'Datenträger einlegen' wenn ein USB-Laufwerk ohne eingelegtes Medium angeschlossen wurde
- Neu/Fix: Dienst-Name geändert von 'USB Drive Letter Manager' in 'USBDLM'

V2.4.2 (16.6.2006)

- Bugfix: bei V2.4.1 konnte der Dienst auf manchen Rechnern beim Systemstart nicht gestartet werden
- Log-Datei zum Debuggen kann aktiviert werden
- kann als 'Eject-Server' für das Kommandozeilen-Tool EjectMedia dienen

V2.4.1 (11.6.2006)

- USB-Floppy-Laufwerke werden ignoriert
- Problem mit doppelt angelegten Mountpoints für Multislot-Cardreader behoben
- Tipp-Fehler im Dienst-Namen korrigiert ('USB drive letter mana**n**ger')
- Errorlevel nach -install, -silentinstall usw. korrigiert

V2.4 (21.5.2006)

- erkennt neue USB-Laufwerk auch dann, wenn sie keinen Mountpunkt bekommen haben damit funktioniert auch das Mounten in NTFS-Ordner vernünftig
- auf Wunsch Löschen von als Mountpunkt benutzten NTFS-Ordnern beim Entfernen des Laufwerks
- Mounten in NTFS-Ordner anhand des Geräte- oder Laufwerks-Namens
- Laufwerksbuchstaben für USB-CD/DVD-Laufwerke

V2.3 (5.5.2006)

• erkennt Netz- und Subst-Laufwerke des angemeldeten Nutzers und remountet ggf. auf den nächsten wirklich freien Buchstaben

V2.2.2 (18.4.2006)

 funktioniert auch mit Laufwerken die nach dem Anschließen mehr als 10 Sekunden brauchen bis der Laufwerksbuchstabe erscheint

V2.2.1 (2.4.2006)

• funktioniert wieder unter Windows 2000 (V2.2 tat das nicht...)

V2.2 (29.3.2006)

- funktioniert unter Windows Vista
- Suche nach c't USB Agent standardmäßig deaktiviert

V2.1 (19.3.2006)

- Vergabe der LW-Buchstaben abhängig vom USB-Port funktioniert auch mit USB-Laufwerken die als 'Lokale Datenträger' eingebunden werden
- Ummounten von Multislot-Cardreadern auch auf Laufwerks-Buchstaben oder Mountpoints

V2.0 (7.3.2006)

- Vergabe der LW-Buchstaben abhängig vom USB-Port
- Ummounten von Multislot-Cardreadern in einen NTFS-Ordner
- Statt Laufwerks-Buchstaben sind auch leere NTFS-Ordner als Mountpoints möglich (experimentell)

V1.1.1 (17.1.2006)

• Fehler bei der Erkennung von USB und Firewire behoben der eine 'Access Violation' verursachte

V1.1 (15.1.2006)

- Einige Timeouts verlängert
- Erkennung von USB und Firewire funktioniert auch unter Windows 2000

V1.0 (10.1.2006)

- Vergabe der LW-Buchstaben abhängig von der Datenträger-Größe möglich
- Ausschluß-Liste für LW-Buchstaben, um eine USBDLM.INI auf dem angeschlossenen Datenträger
- zu 'überstimmen'
- Ausschluß-Liste für Datenträger-Bezeichnungen
- Kein Ummounten, wenn auf dem Laufwerk eine usbagent.inf liegt und der c't USB Agent läuft
- Nur noch für Nutzung im privaten und Bildungs-Bereich Freeware

V0.6 (15.12.2005)

- Vorhandensein eines Mediums im Laufwerk wird nicht mehr an dessen Seriennummer festgemacht, da
- sich gezeigt hat, dass Sticks ab Werk oft keine haben

V0.5 (24.10.2005)

• Kümmert sich nur noch um Laufwerke, die Typ 'DRIVE_FIXED' oder 'DRIVE_REMOVABLE' sind

und

• wechselbare Medien haben oder Hotplug-Geräte sind. Letzterer Test funktioniert nur unter XP...

V0.4 (5.9.2005)

• Kümmert sich auch um Laufwerke mit Typ 'DRIVE_FIXED' (bisher nur 'DRIVE_REMOVABLE')

V0.3 (25.8.2005)

• Erste veröffentlichte Version

Lizenz-Vereinbarung

Software-Lizenz USBDLM, Stand 16. Oktober 2007

1. Gegenstand des Vertrages

Gegenstand des Vertrages ist die beiliegende Software, die Programmbeschreibung und Bedienungsanleitung, sowie sonstiges zugehöriges schriftliches Material, im folgenden auch als Software bezeichnet. Der Autor macht darauf aufmerksam, dass es nach dem Stand der Technik nicht möglich ist, Computersoftware so zu erstellen, dass sie in allen Anwendungen und Kombinationen fehlerfrei arbeitet. Gegenstand des Vertrages ist daher nur eine Software, die im Sinne der Programmbeschreibung und der Benutzungsanleitung grundsätzlich brauchbar ist.

2. Lizenzgewährung

Der Autor gewährt jedem interessierten Anwender im privaten Umfeld, öffentlichen Bibliotheken deren Nutzung kostenlos ist, sowie Bildungseinrichtungen eine kostenlose Nutzungslizenz. Eine Bildungseinrichtung gilt als solche, wenn Schüler und Studenten gegenüber sonstigen Mitarbeitern in der Mehrheit sind. Es ist dann keine explizite Lizenzerteilung erforderlich.

Sonstigen Anwendern wird eine Testlizenz für einen Testzeitraum von 30 Kalendertagen eingeräumt. Nach Ablauf dieser Testlizenz muss der Anwender entweder eine Nutzungslizenz pro Computer erwerben oder die Software muss von den betreffenden Computern entfernt werden.

Der Lizenznehmer erhält das nicht ausschließliche, unwiderrufliche, übertragbare, zeitlich und räumlich unbeschränkte Recht, die Software auf der entsprechenden Anzahl von Computern innerhalb seiner Firma bzw. Organisation einzusetzen.

Die Nutzungslizenz gilt für alle Versionsnummern der Software jetzt und in Zukunft für die entsprechende Anzahl von Lizenzen. Der Lizenznehmer hat kein Recht auf Updateversorgung durch den Autor, da aktuelle Versionen zum Download stehen.

Überprüfbarer Beleg für die Lizenz ist die überlassene Lizenznummer. Diese wird für den Betrieb der Software nicht benötigt. Die Software hat keine eingebauten Beschränkungen und läuft nicht ab.

Die Software kann frei kopiert und weitergegeben werden, soweit folgende Bestimmungen beachtet werden: Die Software darf nur in den vom Autor erstellten MSI- oder ZIP-Archiven weitergegeben werden. Keine Datei darf im Inhalt irgendwie verändert werden, außer für internen Gebrauch.

3. Keine weiteren Rechtsansprüche

Der Autor ist sowohl Urheber als auch Eigentümer der Software, sowie in dieser Software verwendeter Algorithmen und Verfahren. Der Autor behält den Rechts- und Besitzanspruch an der Software.

4. Keine Modifikationen oder Dekompilierung

Es ist nicht gestattet, die Software oder Kopien im Ganzen oder teilweise zu verändern, zu übersetzen, zu disassemblieren oder zu dekompilieren.

5. Redistribution

Redistribution der unveränderten Software-Archivdateien auf CD/DVD von Computer-Zeitschriften und sonstigen Publikationen ist gestattet. Zum Download bitte auf http://www.uwe-sieber.de/files/usbdlm.zip verweisen.

6. Haftungsausschluss-Erklärung

Soweit gesetzlich zulässig, haftet der Autor oder der Händler auf keinen Fall für irgendwelche Schäden aus entgangenem Gewinn, Betriebsunterbrechung, Datenverlust, für Schäden an Hardware oder anderer Schäden gleich welcher Art, die aus der Benutzung dieser Software oder aus der Tatsache, dass sie nicht benutzt werden kann, resultieren, selbst wenn der Autor oder der Händler auf die Möglichkeit solcher Schäden hingewiesen oder unterrichtet wurden. In jedem Fall ist die gesamte Haftung seitens des Autors oder des Händlers unter jedweder Bestimmung begrenzt auf die Summe, die von Ihnen tatsächlich für diese Softwarelizenz bezahlt worden ist. Der Autor haftet nicht für Schäden, es sei denn, dass ein Schaden durch Vorsatz seitens des Autors verursacht worden ist. Gegenüber Kaufleuten wird auch die Haftung für grobe Fahrlässigkeit ausgeschlossen. Eine Haftung wegen evtl. vom Autor zugesicherten Eigenschaften bleibt unberührt. Eine Haftung für

Mangelfolgeschäden, die nicht von der Zusicherung umfasst sind, ist ausgeschlossen.

7. Gewährleistung

Der Autor übernimmt keine Haftung für die Fehlerfreiheit der Software. Insbesondere übernimmt der Autor keine Gewähr dafür, dass die Software den Anforderungen und Zwecken des Erwerbers genügt oder mit anderen von ihm ausgewählten Programmen zusammenarbeitet. Die Verantwortung für die richtige Auswahl und die Folgen der Benutzung der Software, sowie der damit beabsichtigten oder erzielten Ergebnisse, trägt der Erwerber. Das gleiche gilt für das die Software begleitende schriftliche Material.

8. Gesetzesanwendung

Dieser Vertrag unterliegt der Gesetzgebung der Bundesrepublik Deutschland und ist so anzuwenden, als ob die betreffenden Rechtsgeschäfte in der Bundesrepublik Deutschland stattfänden.

9. Auflagen

Der Autor hat das Recht, den Vertrag und die Lizenzrechte sofort zu kündigen, falls Sie die Bestimmungen dieses Vertrages nicht erfüllen.

Konfiguration

USBDLM wird über eine INI-Datei namens USBDLM.INI konfiguriert die im selben Verzeichnis wie die USBDLM.EXE liegt. Der Windows-Explorer zeigt ja standardmäßig keine Dateierweiterungen wie "INI" an - INI-Dateien nennt er "Konfigurationseinstellungen".

Das Editieren der USBDLM.INI erfordert Admin-Rechte, da sie im Programme-Verzeichnis liegt - da haben auf NTFS-formatierten Laufwerken nur Admins Schreibzugriff.

Unter Vista und Windows 7/8 kann hier der "Virtual Store" zuschlagen: Das Schreiben der INI-Datei wird dann nach

C:\Users\(UserName)\AppData\Local\VirtualStore\Program Files\USBDLM umgelenkt. Der USBDLM-Dienst liest aber aus seinem Verzeichnis. Also unter Vista die USDBLM.INI immer als "echter" Administrator bearbeiten - genau dafür sorgt die _edit-ini.cmd.

Oder USBDLM nicht unter C:\Programme installieren, sondern irgendwo anders, z.B. C:\Tools\USBDLM - der Virtual Store ist nur für C:\Programme wirksam.

Die INI am besten immer über die _edit-ini.cmd öffnen - die fordert falls erforderlich die nötigen Admin-Rechte an.

Falls noch keine USBDLM.INI vorhanden ist legt _edit-ini.cmd eine solche mit grundlegenden Einträgen an.

Wie INI-Dateien funktionieren

INI-Dateien sind in Abschnitte unterteilt - jeder Abschnitt beginnt mit dem Abschnitts-Namen in eckigen Klammern, z.B. [DriveLetters] und erstreckt sich bis zum Beginn des nächsten Abschnitts bzw. beim letzten Abschnitt bis zum Datei-Ende.

Die Konfigurations-Einstellungen stehen unter den jeweiligen Abschnitts-Namen - ihre Bedeutung ist abhängig vom Abschnitt in dem sie stehen.

Kommentarzeilen beginnen mit einem Semikolon. Kommentare sind nur am Zeilenanfang zulässig.

In normalen INI-Dateien darf es jeden Abschnitt und jeden Wert innerhalb eines Abschnitts nur einmal geben. Ab USBDLM V5.0 besteht diese Beschränkung nicht mehr.

Niemals nur Abschnitts-Zeilen so auskommentieren:

;[DriveLetters] DriveLetter=X-Z

Die nicht auskommentierten Einstellungen aus diesem Abschnitt gehören dann zum vorherigen! So legt man einen Abschnitt z.B. korrekt lahm:

[----DriveLetters]

DriveLetter=X-Z

Es ist nach wie vor ein Abschnitt, aber keiner für den sich jemand interessiert.

Alternativ Registry

Ab V3.3.1 kann USBDLM seine Einstellungen alternativ aus der Registry lesen, und zwar unter HKLM/Software/Uwe Sieber/USBDLM

Wenn dieser Schlüssel vorhanden ist, wird die USBDLM.INI ignoriert! Nur die Einstellungen für die Log-Datei kommen immer aus der USBDLM.INI.

Für jeden INI-Abschnitt ist ein Schlüssel anzulegen, die Werte müssen vom Typ REG_SZ sein. Damit USBDLM Änderungen im laufenden Betrieb bemerkt, kann der Wert (Standard) unter

HKLM/Software/Uwe Sieber/USBDLM auf einen anderen (beliebigen) Wert gesetzt werden. USBDLM liest dann die Konfiguration neu ein.

Hier gibt's eine Beispiel REG-Datei (natürlich erst speichern und bearbeiten, nicht gleich ausführen...): http://www.uwe-sieber.de/files/usbdlm beispiel.reg

Unter x64 Windows wird die Win32-Version von USBDLM umgeleitet nach HKLM/Software/<u>Wow6432Node</u>/Uwe Sieber/USBDLM, also besser die x64-Version einsetzen.

In der Registry gilt natürlich weiterhin die Regel, dass es jeden Key nur einmal geben darf, ebenso jeden Wert unterhalb eines Keys.

Einige Beispiele

Ganz ohne Konfiguration behebt USBDLM Konflikte mit Netz- und Subst-Laufwerken.

USBDLM bietet vielfältige Möglichkeiten, auf die Vergabe von Laufwerksbuchstaben Einfluss zu nehmen. Für jede Vorgabe gibt es einen Abschnitt in der USBDLM.INI. Die einfachste Variante für Laufwerksbuchstaben für USB-Sticks und Festplatten sieht so aus:

USBDLM vor V5.0

[DriveLetters]

Letter1=U

Letter2=V

Letter3=W

Letter4=X

Letter5=Y

Ab V5.0 braucht man nicht mehr nummerieren, es geht allein nach der Reihenfolge:

[DriveLetters]

Letter=U

Letter=V

Letter=W

Letter=X

Letter=Y

Die Buchstaben werden von 1 bis 9 eingelesen und anschließend zehn nicht nummerierte. Somit ist USBDLM V5.0 kompatibel mit vorhandenen INI-Dateien, man kann die Nummerierungen jetzt aber weglassen.

Auf den ersten geeigneten Buchstaben wird dann umgemountet.

Wenn nur Buchstaben im Spiel sind, kann man auch eine kompakte Notation in Listenform nutzen:

[DriveLetters] Letters=U,V,W,X,Y

Oder noch kürzer:

[DriveLetters] Letters=UVWXY

Aufeinanderfolgende Buchstaben können per "von-bis" notieret werde:

[DriveLetters] Letters=U-W

U-W wäre die Kurzfassung von U,V,W. Absteigend geht's auch, W-U steht also für W,V,U.

Ab V4.3 verstehen alle Einträge alle Notationsvarianten (Buchstabe, Buchstaben-Liste, Mountpunkt, INI-Datei), eingelesen werden sie in folgender Reihenfolge:

Letters

Letter

Letter1 bis Letter9

Ab V5.0 ist die Reihenfolge

Letters

Letter

Letter1 bis Letter9

Letter

Letter

••

Letter

Also besser nicht gemischt mit und ohne Nummerierung arbeiten, das wird unübersichtlich.

Wenn generell Mangel an Laufwerksbuchstaben herrscht, kann nach dem letzten Buchstaben noch einen NTFS-Ordner für Notfälle (kein freier Buchstabe übrig) konfiguriert werden: Letter=C:\%FriendlyName%

%FriendlyName% ist eine USBDLM-<u>Variable</u> und wird durch den Geräte-Namen des Laufwerks ersetzt. USBDLM legt dann einen Ordner an und mountet das Laufwerk dorthin. Das funktioniert nur auf NTFS-

Laufwerken, das Dateisystem des zu mountenden Laufwerks ist dagegen egal.

USBDLM ordnet standardmäßig immer nur einen Laufwerksbuchstaben oder Mountpunkt zu. Will man Laufwerksbuchstaben und NTFS-Mountpunkt gleichzeitig oder auch mehrere NTFS-Mountpunkte haben, kann ab V4.3 MaxMountPoints auf einen Wert größer 1 konfiguriert werden, entweder im betreffenden Abschnitt oder als Voreinstellung im Abschnitt Settings.

Beispiel für Buchstabe U: oder V: und zwei NTFS-Ordner:

[DriveLetters]

MaxMountPoints=3

Letters=U,V

Letter=C:_USB\%DevName%

Letter=C:_Share\Drive_%DriveLetter%

MaxMountPoints führt aber nicht zum löschen bereits vorhandener, dann u.U. überzähliger Mountpunkte. Nur USBDLM hört auf, MountPunkte anzulegen, wenn MaxMountPoints erreicht ist.

Das Zuordnen mehrerer Buchstaben lässt Windows nicht zu, wenn man unbedingt will, kann man sich mit dem Windows-SUBST-Kommando behelfen.

Kein Buchstabe / Laufwerk entfernen

Um ein Laufwerk zu verstecken, kann auch "kein Buchstabe" konfiguriert werden - ein einzelnes Minus-Zeichen steht dafür.

Beispiel um die Buchstaben aller USB-Laufwerke zu entfernen:

[DriveLetters]

Letter=-

Soll das Gerät gleich ganz entfernt werden ("Vorbereiten zum sicheren Entfernen"), konfiguriert man 2x Minus:

[DriveLetters]

Letter=--

Um auf einer USB-Festplatte die erste Partition auf X: zu haben und die zweite Partition zu verstecken:

[DriveLetters]

Letter=X

Letter=-

Das funktioniert aber nur, wenn X: auch frei ist, sonst wird für alle Partitionen der Platte der Buchstabe entfernt!

Um auf einer USB-Festplatte die erste Partition zu verstecken und die zweite unter X: zu haben, funktioniert das **nicht**:

[DriveLetters]
Letter=Letter=X

Warum? Es wird für das Laufwerk und den Letter-Werten nach einem freien Buchstaben gesucht. "Kein Buchstabe" ist aber immer frei, somit wird die zweite Letter-Zeile nie erreicht. Allen Partitionen wird der Buchstabe weggenommen.

Hier hilft ein eigener Abschnitt mit der Partitionsnummer als Kriterium.

Buchstaben abhängig von bestimmten Kriterien

Um einen [DriveLetters] Abschnitt nur auf bestimmte Laufwerke anzuwenden, kann man Kriterien definieren. Die Abschnitte musste man dann vor V5.0 nummerieren, ab V5.0 geht es bei nicht nummerierten Abschnitten allein nach der Reihenfolge der Abschnitte - es wird aufsteigend nach einem Abschnitt mit passenden Kriterien gesucht.

USBDLM benutzt immer nur die Einstellungen des ersten passenden Abschnitts! Es gibt kein Ausweichen auf den nächsten passenden Abschnitt, falls alle konfigurierten Buchstaben eines Abschnitts vergeben sind!

Beispiel zum Sperren von Laufwerken über 10 GB, kleinere auf U, V und W:

;ab 10GB Buchstaben entfernen [DriveLetters] MinDriveSize=10GB Letter=-

;alle anderen auf U, V, W
[DriveLetters]
Letter=U
Letter=V
Letter=W

Oder kurz:

[DriveLetters] Letters=U,V,W

Das ist ausführlich unter Buchstaben abhängig von bestimmten Kriterien beschrieben.

Mounten in NTFS-Ordner

Statt Laufwerks-Buchstaben können auch leere NTFS-Ordner als Mountpoints ('Bereitstellungspunkte') angegeben werden, siehe <u>Mounten in NTFS-Ordner</u>

Das Anzeigen von Balloontips, das Überprüfen von Laufwerksbuchstaben beim Start und vieles mehr lässt

sich konfigurieren: Sonstige Einstellungen - [Settings]

Einstellungen

Die im Folgenden beschriebenen Einstellungen sind in der USBDLM.INI im Abschnitt [Settings] einzutragen.

In den Beispielen wird immer wieder aufs Neue der Abschnitt [Settings] genannt. Damit ist natürlich der eine gemeinsame Abschnitt gemeint. Auch mit USBDLM V5 kann es nur einen geben...

Die Einstellungen werden ab V4.2 bis auf wenige Ausnahmen sofort wirksam.

Laufwerks-Buchstaben beim Start überprüfen

Die gemachten Laufwerks-Einstellungen werden standardmäßig bei Start des USBDLM-Dienstes überprüft, um auch beim Booten bereits angeschlossene Laufwerke zu erfassen. Soll USBDLM das nicht tun:

[Settings] CheckLettersOnStartup=0

Konflikte mit Netz-Laufwerken können hier nicht ohne weiteres behoben werden, da diese zu diesem Zeitpunkt noch nicht vorhanden sind und USBDLM nicht in die Zukunft blicken kann... Außerdem würde das Anlegen der Netzlaufwerke eh fehlschlagen, wenn der jeweilige Buchstaben schon besetzt ist. Künftige Netz-Laufwerke, die hier berücksichtigt werden sollen, müssen deshalb im Abschnitt [NetworkLetters] eingetragen werden.

Laufwerks-Buchstaben beim Nutzer-Login / Nutzerwechsel überprüfen

Die konfigurierten Laufwerks-Einstellungen werden standardmäßig beim Wechsel des aktiven Nutzers überprüft, um nutzerabhängige Konfigurationen zu berücksichtigen. Soll USBDLM das nicht tun:

[Settings] CheckLettersOnLogon=0

Laufwerks-Buchstaben beim Aufwachen aus Standby und Ruhezustand überprüfen

Bei manchen Windows-Installationen kommen nach dem Aufwachen aus Standby und Ruhezustand ("Resume") die Laufwerksbuchstaben durcheinander. USBDLM kann dann die Buchstaben überprüfen, normalerweise ist das aber nicht nötig.

Soll USBDLM beim Resume die Buchstaben überprüfen:

[Settings]

AutoRuns bei Nutzer-Login / Nutzerwechsel ausführen

Die konfigurierten OnArrival-Ereignisse werden standardmäßig nur beim Anschließen eines Laufwerks bzw. Einlegen eines Datenträgers überprüft und ggf. ausgeführt.

Soll USBDLM AutoRuns auch beim Login oder Nutzerwechsel ausführen:

[Settings]

AutoRunOnLogon=1

Beim automatischen Logon ist der USBDLM-Dienst möglicherweise noch nicht gestartet, da XP nicht auf den Start aller Dienste wartet - die Illusion vom schnellen Windows-Start hat Vorrang. USBDLM verpasst so das Logon-Ereignis.

Deshalb führt USBDLM beim Dienst-Start die AutoRuns dennoch aus, wenn seit dem Systemstart weniger als zwei Minuten vergangen sind und bereits ein Nutzer eingeloggt ist, es sich also offenbar um einen verpassten Logon handelt.

Die Zeitspanne innerhalb der USBDLM von einem verpassten Logon ausgeht kann in Millisekunden konfiguriert werden:

[Settings]

AssumeMissedLogonMaxTime=120000

Sichere Kommandozeilen

Bei Kommandozeilen die Leerzeichen enthalten und die ausführbare Datei nicht in Anführungszeichen eingeschlossen ist, muss USBDLM raten, wo die ausführbare Datei endet und die Parameter beginnen. Das funktioniert zwar, ist aber unsicher.

Beispiel:

open=C:\Program Files\tool.exe

Hier kann als ausführbare Datei C:\Program Files\tool.exe ohne Parameter gemeint sein, oder aber C:\Program.exe mit Files\tool.exe als Parameter.

USBDLM würde zwar C:\Program Files\tool.exe ausführen, aber es gibt schwierigere Fälle wo es auch mal daneben gehen könnte.

Mit SafeCommandLines=1 akzeptiert USBDLM keine Kommandozeilen, die Leerzeichen enthalten, die ausführbare Datei aber nicht in Anführungsstriche eingeschlossen ist:

Beispiele für ungültige Kommandozeilen:

open=C:\Program Files\tool.exe

open=C:\Tools\tool.exe -x

Obige Kommandozeilen gültig:

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

open="C:\Tools\tool.exe" -x

Default ist 0 um Probleme mit vorhandenen Konfigurationen zu vermeiden, in der von _edit_ini.cmd

angelegten USBDLM.ini steht's für neue Nutzer aber auf 1:

[Settings]

SafeCommandLines=1

UsbDriveInfo sucht nach problematischen Kommandozeilen und zeigt einen Fehler oder eine Warnung, je nach dem wie SafeCommandLines steht.

USBDLM warnt mit einem Balloon-Tip.

NTFS-Ordner beim Entfernen des Laufwerks löschen

Wird in Ordner auf <u>NTFS-Laufwerken gemountet</u>, werden diese Ordner beim Entfernen des dort bereitgestellten Laufwerks durch USBDLM gelöscht. Wenn USBDLM das nicht tun soll:

[Settings]

DeleteMountPointsOnRemoval=0

Richtlinie "Für Leistung optimieren" für USB-Laufwerke

Unter XP und höher kann man für USB-Laufwerke zwischen den "Richtlinien" "Für schnelles Entfernen optimieren" und "Für Leistung optimieren" wählen, mit der Behauptung, letztere würde einen Schreibcache aktivieren. Tut es aber nicht, wenn es sich beim Laufwerk um einen USB-Wechseldatenträger handelt - so geben sich fast alle USB-Sticks zu erkennen! "Für Leistung optimieren" aktiviert für Wechseldatenträger lediglich die Möglichkeit, das Laufwerk mit NTFS zu formatieren und dann bekommt man tatsächlich einen Schreibecache - auch auf anderen Rechnern, da unabhängig von der Richtlinie... Es geht allein um die Erlaubnis, mit NTFS zu formatieren.

Auf FAT-formatierten USB-Festplatten wird tatsächlich ein Schreibcache aktiviert, mit NTFS hat man auch hier in jedem Fall einen Schreibcache.

USBDLM kann die Richtlinie für alle USB-Laufwerke auf "Für Leistung optimieren" umstellen:

[Settings]

OptimizeUsbDrivesForSpeed=1

Bei erstmals angeschlossenen Laufwerken klappt das sofort. Vorhandene Laufwerke müssen zweimal neu angeschlossen werden (beim ersten Mal wird der Registry-Eintrag gemacht, beim zweiten Mal wird er wirksam).

Nebenwirkung: Der Windows 10 Task-Manager zeigt Wechseldatenträger auf seinem Reiter "Leistung" wenn sie für Leistung optimiert sind.

Schreibcache für USB-Laufwerke erzwingen

Unter XP und höher haben FAT-formatierte USB-Laufwerke standardmäßig keinen Schreibcache - das Schreiben vieler kleiner Dateien geht entsprechend langsam und stresst das Laufwerk. Mit der Richtlinie "Für Leistung optimieren" kann man den Schreibcache für USB-Festplatten (DriveType

FIXED) aktivieren, bei USB-Wechseldatenträgern (DriveType REMOVABLE) ist das dagegen völlig wirkungslos, siehe vorherige Abschnitt.

Es gibt aber einen Wert in der Registry, mit dem sich der Schreibcache erzwingen lässt, auch für USB-Wechseldatenträger (WriteCacheEnableOverride). Setzt man diesen für ein Laufwerk auf 1 bekommt es einen Schreibcache spendiert.

Um Datenverluste zu vermeiden, muss man dann natürlich immer schön "Hardware sicher entfernen" oder ein anderes Tool, wie mein <u>RemoveDrive</u> oder <u>EjectMedia</u> benutzen.

USBDLM kann diesen Wert für alle USB-Laufwerke setzen:

[Settings]

UsbWriteCacheEnable=1

Bei erstmals angeschlossenen Laufwerken klappt das sofort. Vorhandene Laufwerke müssen zweimal neu angeschlossen werden (beim ersten Mal wird der Registry-Eintrag gemacht, beim zweiten Mal wird er wirksam).

- UsbWriteCacheEnable=0 -> setzt WriteCacheEnableOverride auf 0
- UsbWriteCacheEnable=1 -> setzt WriteCacheEnableOverride auf 1
- UsbWriteCacheEnable=-1 -> WriteCacheEnableOverride wird nicht verändert (default)

Datenträgerverwaltung automatisch starten für leere Festplatten

Beim Anschließen einer komplett leeren Festplatte gibt's mangels Partition keinen Laufwerksbuchstaben. Die Platte muss in der Datenträgerverwaltung "initialisiert", partitioniert und formatiert werden. USBDLM startet die Datenträgerverwaltung automatisch, wenn eine USB-Festplatte mit komplett leerem Bootsektor angeschlossen wird:

[Settings]

StartDiskMgmtForUninitializedDrives=1

- 0: deaktiviert
- 1: für USB-Festplatten (default)
- 2: für alle Festplatten

Auf externe Mount-Ereignisse reagieren

Wenn eine andere Software einen Laufwerksbuchstaben ändert, bekommt das ein Dienst wie USBDLM nicht gemeldet. Um sowas dennoch mitzubekommen, erzeugt USBDLM ein unsichtbares Fenster, das diese Benachrichtigung empfängt.

Unter Vista, Win7 und unter XP bei mehr als einem angemeldeten Nutzer reicht das nicht, hier startet USBDLM die USBDLM_usr.exe im Kontext des Nutzers zum Empfang der Benachrichtigungen.

Wenn man das alles nicht braucht:

[Settings] ExternalMountEvents=0

Am unsichtbaren Fenster bzw. der USBDLM_usr.exe hängen auch die OnArrival-Events und Balloontips für gemountete TrueCrypt-, Subst- und Netzlaufwerke - die bekommt USBDLM auch nur so gemeldet. Wenn davon etwas genutzt wird, ist ExternalMountEvents=0 wirkungslos.

Standardmäßig ermittelt USBDLM anhand der Einstellungen ExternalMountEvents, [Balloons] ShowVirtual und der verwendeten OnArrival/OnRemoval-Abschnitte, ob USBDLM_usr.exe benötigt wird. Das Laden der USBDLM_usr.exe kann auch gezielt gesteuert werden, siehe auch <u>Technische Infos</u>:

[Settings] LoadUsbdlm_Usr=0

- 0: nicht laden
- 1: laden falls wie benötigt (default)
- 2: immer laden (bei aktiver UAC für den eingeschränkten Nutzer)
- 3: immer laden (bei aktiver UAC für den eingeschränkten und für elevated Nutzer)

Volume-Infos Timeout

Beim Lesen einer Datei und beim Ermitteln von Volume-Label, Volume-Serial und Dateisystemname wartet USBDLM maximal 10 Sekunden auf das Ergebnis. Wenn länger gewartet werden soll, kann das in Millisekunden festgelegt werden:

[Settings] VolumeInformationMaxWait=10000

Debug-Infos

Wenn es Probleme gibt, kann man u.U. anhand von Debug-Ausgaben herausfinden, was schiefläuft. Für Ausgaben in die Log-Datei ist WriteLogFile=1 zu setzen. Standardmäßig wird eine _USBDLM.LOG ins Stammverzeichnis des Windows-Laufwerks geschrieben. Mit LogFile=Pfad\Datei lässt sich ein anderes Ziel festlegen.

Für Live-Debug-Ausgaben, die man mit Tools wie <u>SysInternals DebugView</u> oder <u>TraceSpy</u> einfangen kann, ist WriteDebugOutput=1 zu setzen.

Wie detailliert die Ausgaben ausfallen, wird über den LogLevel 1 bis 5 gesteuert, wobei es ab 4 extrem ins Detail geht. Beispiel:

[Settings]
LogLevel=3
WriteLogFile=1
LogFile=C:\USBDLM.LOG
WriteDebugOutput=1

LogLevel=2 reicht, um zu sehen, wie die Abschnitte aus der INI für ein Laufwerk ausgewählt werden. Mit LogLevel=3 wird's dann unübersichtlich und LogLevel=4 bringt detaillierte Ablauf-Infos, die nur dem Autor bei der Fehlersuche helfen.

LogLevel=0 schreibt nur je eine Zeile beim Anschließen und Entfernen eines Laufwerks, Ausführen einer open= Zeile und beim Starten und Beenden des USBDLM-Dienstes.

Mit LogShowDate=1 wird jeder Log-Zeile das Datum vorangestellt, mit LogShowTech=0 werden Timing, Loglevel und Thread-ID weggelassen.

Ab V4.2 ist dafür kein Neustart des USBDLM-Dienstes nötig. Die Einstellungen für die Log-Datei werde immer aus der INI gelesen, auch wenn die anderen Einstellungen aus Registry kommen sollten.

Beim Neustart des USBDLM-Dienstes wird die Logdatei gelöscht. Damit dabei potentiell wertvolle Logs nicht verloren gehen, können Kopien angelegt werden, Standard ist zwei: LogFileBakCount=2

So wird beim Start USBDLM_BAK2.LOG gelöscht, USBDLM_BAK1.LOG in USBDLM_BAK2.LOG umbenannt und USBDLM.LOG in USBDLM_BAK1.LOG.

Wenn man das Datum am Anfang jeder Log-Zeile will: LogShowDate=1

Cache flushen und Dateisystem dismounten bei Standby und Ruhezustand

Wenn Windows in Standby oder Ruhezustand geht, flusht USBDLM den Schreibcache aller Laufwerke. Die Dateisysteme bleiben aber gemountet, deshalb ist es keine gute Idee dann ein anderes Betriebssystem zu laden, das dann auch die Dateisysteme mountet - das führt zu Inkonsistenzen, ein Dateisystem kann nicht Diener zweier Herren sein.

USBDLM kann versuchen, alle Laufwerke (außer dem Windows-Laufwerk) zu dismounten:

[Settings]

DismountOnSleep=1

Das Dismounten klappt aber nur, wenn keine offene Dateien auf dem jeweiligen Laufwerk sind. Das Dismounten kann per DismountOnSleep=2 für USB und andere Hotplug-Laufwerke erzwungen werden, offene Datei-Handles werden dann ungültig, die zugehörige Software wird sich dann beim Wiederaufwachen beschweren oder merkwürdig reagieren. Mit DismountOnSleep=3 wird es für alle Laufwerke erzwungen.

Das System-Volume dismountet USBDLM nicht. Ich habe auch nicht getestet, was passiert, wenn man es erzwingt.

Ab V5.1 prüft USBDLM zunächst, ob ein Laufwerk schläft und läßt es dann in Ruhe außer DismountOnSleep=4.

Entfernung von Schreibrechten für Nicht-Admins auf USBDLM.EXE und USBDLM.INI

Beim Start überprüft USBLDM, ob Nicht-Admins Schreibzugriff auf USBDLM.EXE und USBDLM.INI haben. Wenn dem so ist, ändert USBDLM die Zugriffsrechte entsprechend. Wenn man das nicht will:

[Settings]

Kommandoschnittstelle für Programme die mit eingeschränkten Rechten arbeiten

Einige nützliche Windows-Funktionen benötigen Admin-Rechte, obwohl sie völlig harmlos sind, z.B. das Flushen des Schreibcaches von Datenträgern.

Ab V4.8 hat USBDLM dafür eine Kommando-Schnittstelle, die einige Funktionen zur Verfügung stellt. Welche freigeschaltet sind, kann konfiguriert werden, die Funktionen sind bit-kodiert:

Werte für die Funktionen, bei Bedarf addieren:

Keine Funktion0 (default)Schreibcache flushen1Dismounten2Auswerfen4

Auswerfen oder Dismounten erzwingen 8 Hardware sicher entfernen: 16 Volume on/offline nehmen 32 Code-21-Geräte reaktivieren 64

Bei Bedarf addieren. Beispiel für alles außer "erzwingen": (1+2+4+16+32+64):

[Settings]

CommandInterface=119

Beispiel für alles: (1+2+4+8+16+32+64):

[Settings]

CommandInterface=127

255, 65535, 0xFF usw. gehen auch.

Ein Laufwerk zu sperren heißt, sich exklusiven Zugriff zu verschaffen. Das klappt nur wenn sonst niemand auf das zu sperrende Laufwerk zugreift, sprich, keine Dateien geöffnet sind. Sperren ist die Vorraussetzung, um ein Laufwerk korrekt zu dismounten. Ohne Sperren geht's auch, offene Dateien bleiben aber offen, Zugriffe führen aber zur einer Fehlermeldung, die betroffene Software wird nicht gut drauf reagieren.

Tools die die Funktionen benutzen:

- FFB: Cache flushen und ggf. dismounten
- <u>EjectMedia</u>: Cache flushen, dismounten, auswerfen, ggf. mit erzwingen, Volume "offline" nehmen
- <u>LoadMedia</u>: Auswerfen rückgängig, Volume "online" bringen
- RemoveDrive: Cache flushen, dismounten+auswerfen, reaktivierbares "Hardware sicher entfernen"
- RestartSrDev: Geräte mit Problemcode 21 und u.U. 47 reaktivieren

Reservierte Buchstaben

Alle im Abschnitt [NetworkLetters] konfigurierten Buchstaben sind für die Benutzung durch USB-Laufwerke gesperrt, das heißt, wenn Windows einen dieser Buchstaben einem Laufwerk zuordnet, wird USBDLM eingreifen und einen anderen Buchstaben zuordnen, auch wenn keine für das Laufwerk zuständiger Abschnitt gefunden wird oder alle dort konfigurierten Buchstaben schon belegt sind.

Wenn überhaupt, sollten hier nur Laufwerksbuchstaben von Netz- und Subst-Laufwerken eingetragen werden, damit USBDLM sie schon beim Systemstart berücksichtigen kann. Außerdem können Buchstaben konfiguriert werden, die nicht über die normalen Mount-Mechanismen zugeordnet werden, also nicht in der Datenträgerveraltung erscheinen, z.B. TrueCrypt-Laufwerke.

Niemals Buchstaben von lokalen Laufwerken konfigurieren - USBDLM würde sie ummounten, Hauptsache weg von den NetworkLetters!

Bis V4.2 hieß der Abschnitt [ExcludedLetters], was so auch weiterhin funktioniert. Der Abschnitt wurde umbenannt, weil viele Nutzer hier die Buchstaben ihrer internen Laufwerke konfiguriert haben, was falsch ist und zu ungewollten Ergebnissen führt.

Alternative: Einfach in den DriveLetters-Abschnitten ausreichend Buchstaben konfigurieren, so dass die Buchstaben der Netz- und Subst-Laufwerke von selbst frei bleiben.

Die Liste überstimmt auch die Konfiguration von Buchstaben in externen INI-Dateien, die per Letters=%drive%\usbdlm.ini eingelesen werden (Das hat von V4.7 bis einschl. 4.8.8.1 nicht funktioniert).

Folgendes Beispiel sperrt F und G für USB-Laufwerke:

[NetworkLetters] Letters=F,G

In diesem Abschnitt sind nur Buchstaben, also keine Ordner zulässig!

Standard-Buchstaben für USB-Sticks und Festplatten

In bis zu 100 Abschnitten [DriveLetters] werden Standard-Buchstaben für USB-Festplatten und Sticks festgelegt.

Vor USBDLM V5 mussten die Abschnitte nummeriert werden (jeden durfte es nur einmal geben), also [DriveLetters1] bis [DriveLetters99].
Konfiguriert man über die Registry statt über die INI-Datei, gilt das auch für USBDLM V5.
Beispiel:
[DriveLetters] Letter1=U Letter2=V Letter3=W
Oder als Liste:
[DriveLetters] Letters=U,V,W
oder ganz kurz:
[DriveLetters] Letters=UVW
Da im Abschnitt keine <u>Kriterien</u> angegeben sind, kommen die Standard-Kriterien zum Zuge, das sind
BusType=USB
DriveTypes=REMOVABLE, FIXED
Der Abschnitt ist also für USB-Wechseldatenträger und USB-Festplatten zutreffend.
Die Laufwerksbuchstaben können auch aus einer externen INI-Datei gelesen werden, z.B. auf dem angeschlossenen Laufwerk:
[DriveLetters] Letters=%drive%\usbdlm.ini
Man kann auch beides kombinieren - die externe Datei hat im Beispiel Vorrang, weil sie oben steht:
[DriveLetters]

Letters=%drive%\usbdlm.ini Letters=U,V

In der externen INI-Datei ist nur ein einfacher Abschnitt [DriveLetters] ohne Kriterien zulässig, wie der im ersten Beispiel.

Soll ein Abschnitt nur zuständig sein, wenn die USBDLM.INI auf dem Laufwerk vorhanden ist, braucht's ein entsprechendes Kriterium:

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

Um nach einer Datei zu schauen, muss aber das Dateisystem gemountet sein, das führt auf manchen Laufwerken zu Verzögerungen oder es klemmt ganz.

Priorität

Zuerst kommt Letters zum Zuge, dann Letter und zuletzt Letter1 bis Letter9.

Anzahl begrenzen

Wenn die konfigurierten Buchstaben mit einer Gruppenrichtlinie versehen sind, die z.B. das Ausführen von Programmen verhindert, sollte man sicherstellen, dass das nicht durch Fluten mit Laufwerken ausgehebelt wird, z.B. durch Anschließen eines Multislot-Cardreaders mit 5 Laufwerken. Wenn die im zuständigen DriveLetters-Abschnitt konfigurierten Buchstaben aufgebraucht sind, ordnet USBDLM den ersten freien zu - dort greift dann u.U. die vorgesehene Gruppenrichtlinie nicht.

Das kann man verhindern, indem man als letzten Buchstaben "kein Buchstabe" konfiguriert, dafür steht ein einzelnes Minus-Zeichen:

; maximal 3 USB-Laufwerke [DriveLetters] Letter1=U Letter2=V Letter3=W Letter4=-

Als Liste:

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

Unveränderliche Buchstaben

Alle im Abschnitt [StaticLetters] konfigurierten Buchstaben rührt USBDLM nicht an. So kann man trotz installiertem USBDLM noch bestimmte Laufwerksbuchstaben per Windows Datenträgerverwaltung zuordnen ohne dass USBDLM das beim nächsten Anschließen wieder rückgängig macht.

Beispiel: Laufwerksbuchstaben U bis Z nicht anrühren:

[StaticLetters] Letters=U-Z

Hier nur "hohe" Buchstaben konfigurieren. Buchstaben weiter unten sind ja die die Windows automatisch vergibt und die würde USBDLM dann nicht ändern.

Kein Medium, kein Buchstabe

Der typische 20 in 1 Cardreader verbrät für jeden Karten-Slot einen kostbaren Laufwerks-Buchstaben - ob man eine Karte dafür besitzt oder nicht. Und man weiß eh nie, welcher Buchstabe zu welchem Slot gehört. Also weg mit den Buchstaben und genau dann einen zuordnen, wenn ein Medium eingelegt wird:

```
[Settings]
NoMediaNoLetter=1
```

Um wirksam zu werden, muss das betreffende Gerät neu angeschlossen oder der USBDLM-Dienst neu gestartet werden.

Unterstützt werden standardmäßig nur Wechseldatenträger, aber keine Floppys oder CD-ROMs. Außerdem sind Geräte ausgeschlossen, die "Iomega" und "ZIP" oder "JAZ" enthalten, da für solche floppy-artigen Laufwerke NoMediaNoLetter nicht angemessen wäre. Mit echten Floppy-Laufwerken funktioniert das Ganze überhaupt nicht, das diese keine Benachrichtigungen beim Einlegen und Entfernen einer Diskette generieren.

```
NoMediaNoLetter=0 -> kein Laufwerk
NoMediaNoLetter=1 -> nur Multislot-Cardreader
NoMediaNoLetter=2 -> auch Singleslot-Cardreader
NoMediaNoLetter=3 -> alle Wechseldatenträger
```

Singleslot-Cardreader lassen sich nicht sicher erkennen. USBDLM versucht es anhand einiger charakteristischer Zeichen im Gerätenamen, wie "card", "reader" und einigen anderen. Wenn ein Kartenleser nicht als solcher erkannt wird, entweder NoMediaNoLetter=3 setzten oder einen Abschnitt [NoMediaNoLetterInclude] nutzen, siehe unten.

UsbDriveInfo zeigt, ob NoMediaNoLetter für einen Wechseldatenträger wirksam ist oder nicht. Falls nicht, zeigt es was für NoMediaNoLetter konfiguriert werden muss:

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

Beispiel für ein altes Iomega IDE ZIP-Drive und ein ATAPI-CDROM:

[NoMediaNoLetterInclude]
DeviceID=IDE\DISKIOMEGA_ZIP_100
DeviceID=IDE\CDROMHL-DT-ST_DVDRRW_GWA-4161B

Beim Einlegen eines Mediums ordnet USBDLM dann den ersten freien Buchstaben zu. Soll es ein bestimmter sein, muss man ihn entsprechend konfigurieren:

[DriveLetters]
DeviceID=IDE\DISKIOMEGA_ZIP_100
Letter=Z

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

Geräte von NoMediaNoLetter ausschließen

Mit manchen Geräten gibt es Probleme: USBDLM wird dann mit Nachrichten geflutet oder es nimmt einem Gerät den Buchstaben weg, weil es zunächst wie ohne eingelegten Datenträger erscheint. Anhand der Geräte-ID kann man Geräte ausschließen.

Beispiel zum Ausschließen des Apple iPod:

[NoMediaNoLetterExclude]
DeviceID=ven_apple&prod_ipod

Das ist bereits fest eingebaut, weil die iTunes Software auf den Laufwerksbuchstaben schaut.

Problem: Geräte die nicht starten unter Vista und Windows 7

Für USB-Wechseldatenträger installieren Vista, Windows 7, 8 und überraschenderweise auch Windows 2008 Server (R2) immer auch ein "WPD Dateisystem". WPD steht für "Windows Portable Device", eine Programmierschnittstelle für Geräte wie MP3-Player, Handys u.a.

Wenn ein USB-Wechseldatenträger keinen Laufwerksbuchstaben hat (egal ob USBDLM den entfernt oder ob der Nutzer es manuell erledigt), schlägt der Start des Treibers mit "Code 10" fehl. Macht aber nichts: Als normaler Datenträger funktioniert das Laufwerk trotzdem.

Wenn man den WPD-Kram komplett loswerden will, kann man den Dienst "Enumeratordienst für tragbare Geräte" (WPDBusEnum) deaktivieren. Nebeneffekt: Der Windows Media Player synchronisiert sich dann nicht mehr mit USB-Massenspeichern. Und USB-Geräte die nur WPD verstehen funktionieren natürlich nicht mehr...

Ab Windows 8.1 gibt's das Problem nicht mehr.

Buchstaben kommen immer wieder

Tauchen die Buchstaben hartnäckig immer wieder auf, ist eine andere Software am Werk, die die Buchstaben bei passender Gelegenheit überprüft und ggf. zuordnet.

Bekannt dafür ist das U3 Launchpad auf U3-Sticks (die LaunchU3.exe auf dem Pseudo-CDROM). Weitere Software die grundlos Laufwerksbuchstaben zuordnet ist das "Secure Storage Device SDK" (SSDService.exe) von MXI sowie "TomTom MyDrive Connect".

Mit dem SysInternals ProcessMonitor kann man ermitteln, welcher Prozess Laufwerksbuchstaben zuordnet: http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx

Folgende Filter sollte man setzten:

Path contains DosDevices Operation is RegSetValue

Dann in der Werkzeugleiste rechts den Registry-Button gedrückt lassen, die anderen drei kann man abwählen

Wenn die Buchstaben dann wieder aufgetaucht sind, sollten kurze Zeit später Zugriffe protokolliert worden sein.

Falls das U3-Launchpad schuld ist, bieten sich folgende Workarounds an:

Workaround 1: U3 lahmlegen, indem man dem gefälschten CDROM den Buchstaben wegnimmt:

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

Workaround 2: Keine Admin-Rechte für das U3-Launchpad

Ohne Admin-Rechte kann es keine Laufwerksbuchstaben ändern. Entweder als "Eingeschränkter Nutzer" arbeiten, oder das U3-Launchpad ohne Admin-Rechte starten - das kann USBDLM erledigen:

[OnArrival]
DriveType=CDROM
FileExists=%drive%\u3launch.exe
open=%drive%\u3launch.exe
restricted=1

Oder falls lokal installiert, aus dem Windows Autostart entfernen und durch USBDLM starten lassen:

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

Die aktuelle Version 1.6 des U3-Launchpads funktioniert meist nicht ohne Admin-Rechte und kommt mit einer unsinnigen Fehlermeldung. Hier hilft es manchmal, alle USB-Wechseldatenträger zu entfernen und das U3-Launchpad mit Admin-Rechten auszuführen. Es kann dann seine merkwürdige Installations-Orgie durchziehen und ab dann hoffentlich auch ohne Admin-Rechte arbeiten.

Den Daten-Teil sollte man übrigens nicht auf A:, B: oder gar einen NTFS-Ordner legen, damit ist U3-Launchpad auch überfordert.

Workaround 3: USBDLM dagegen "kämpfen" lassen

Eigentlich hatte ich fest damit gerechnet, dass das unsinnige Verhalten der U3-Software irgendwann behoben wird, aber da es die U3-Entwickler auch nach fünf Jahren offenbar nicht in der Lage sind, das eigene Laufwerk von anderen USB-Wechseldatenträgern zu unterscheiden, muss USBDLM es eben richten:

[Settings]
FightU3Bug=1

Wenn Laufwerkbuchstaben von USB-Wechseldatenträgern unerwartet zugeordnet werden und innerhalb der vergangenen 15 Sekunden ein U3-CDROM angeschlossen wurde, entfernt USBDLM die Buchstaben sofort wieder.

Im Windows-Explorer bleibt dann oft je eine Laufwerks-Leiche mit rotem Fragezeichen zurück. Mit F5 aktualisieren schafft Abhilfe.

Das U3 Launchpad kann sich auch auf das Systemlaufwerk kopieren (unter "All Users\U3") und in den Autostart eintragen. Bei jedem Anschließen eines USB-Wechseldatenträgers ordnet es einen Laufwerksbuchstaben zu, falls nicht vorhanden. Hier hilft dann nur:

[Settings]
ForceNoMediaNoLetter=1

Mounten in NTFS-Ordner

Statt Laufwerks-Buchstaben können auch leere NTFS-Ordner als Mountpoints ('Bereitstellungspunkte') angegeben werden.

Fall nicht vorhanden, legt USBDLM den Ordner, ggf. auch den Pfad an.

Die Ordner bleiben auch nach dem Entfernen des Laufwerks standardmäßig erhalten und können auch problemlos umbenannt werden. USBDLM akzeptiert das, solange der Ordner sich im gleichen übergeordneten Ordner wie der konfigurierte befindet.

Beim Mounten in NTFS-Ordner kann man den Gerätenamen, die Datenträger-Bezeichung oder jede andere <u>USBDLM-Variable</u> nutzen:

Einfach "%DevName%" oder "%VolumeLabel%" konfigurieren - das wird dann durch den Gerätenamen bzw. die Datenträgerbezeichnung ("Volume Label") ersetzt.

Beispiel: Angenommen der Geräte-Name ist "Corsair Flash Voyager"...

[DriveLetters]

Letter1=C:_USB\%DevName%

Dann würde USBDLM den Ordner "C:_USB\Corsair Flash Voyager" anlegen und das Laufwerk dorthin mounten.

Im Gegensatz zu Laufwerksbuchstaben bleiben NTFS-Mountpunkte beim Entfernen des Laufwerks erhalten. USBDLM löscht sie aber standardmäßig, wenn das Laufwerk entfernt wird. Wenn es das nicht tun soll:

[Settings]

DeleteMountPointsOnRemoval=0

Wenn allerdings ein Laufwerk entfernt wird, während der USBDLM-Dienst nicht läuft, z.B. wenn Windows schon heruntergefahren ist, klappt das mit dem Löschen des Ordners natürlich nicht. USBDLM kann das beim nächsten Start oder beim Nutzerwechsel nachholen. Konfiguriert werden die Ordner, die die Mountpunkte enthalten, passend zum oberen Beispiel also

[DeleteUnusedNtfsMountpoints]
Folder1=C:_USB
Folder2=

Ab V4.3 kann USBDLM einem Laufwerk auch mehrere NTFS-Ordner als Mountpunkt zuordnen, auch zusätzlich zu einem Laufwerksbuchstaben. Um kompatibel zu bleiben, ist auch USBDLM V4.3 auf einen Mountpunkt beschränkt, das kann aber über MaxMountPoints=n konfiguriert werden, zum einen global im Abschnitt [Settings] und zum anderen in jedem [DriveLettersXX] Abschnitt.

Beispiel für Buchstabe U: und zwei NTFS-Ordner:

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

Beispiel für Buchstabe U: falls frei, sonst kein Buchstabe, aber immer zwei NTFS-Ordner:

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

Der Standardwert für MaxMountPoints ist 1 und kann im Abschnitt Settings geändert werden. Zulässig sind Werte von 1 bis 10.

[Settings]
MaxMountPoints=10

Das Mounten in NTFS-Ordner bringt allerdings einige **Nachteile** mit sich: Viele Windows-Funktionen für Laufwerke stehen dann über den Windows-Explorer nicht mehr zur Verfügung, z.B. AutoRun, Auswerfen, Formatieren...

Der Windows-Explorer betrachtet das gemountete und das Wirtslaufwerk als ein und dasselbe. Deshalb werden beim Drag+Drop Dateien standardmäßig verschoben statt kopiert.

Für Wechseldatenträger sieht Windows keine Papierkorb-Funktion vor. Mountet man aber einen Wechseldatenträger in einen NTFS-Ordner, wird das von Windows XP beim Löschen nicht berücksichtigt - es wird dann der Papierkorb des Wirts-Laufwerks benutzt! Normalerweise werden die Dateien beim 'Löschen' dorthin verschoben, d.h. es wird nur der Datei-Eintrag geändert, die Daten bleiben unangetastet. Löscht man eine Datei auf einem in ein NTFS-Verzeichnis gemounteten Laufwerk, geht das mit dem Verschieben nicht, da sich Datei und der Papierkorb-Ordner auf verschiedenen Laufwerken und somit in verschiedenen Dateisystemen befinden. Die zu löschende Datei wird deshalb in den Papierkorb kopiert und anschließend gelöscht, was je nach Dateigröße entsprechend dauert. Bei Ordnern funktioniert's unter XP gar nicht - es gibt eine irreführende Fehlermeldung ("Zugriff verweigert"). Also beim Löschen die Umschalt-Taste festhalten um den Papierkorb zu übergehen oder den Papierkorb des Wirtslaufwerks ganz abschalten. Windows 7 macht's besser.

Unter "Hardware sicher entfernen" werden NTFS-Mountpunkte bis einschließlich Windows Vista nicht unterstützt - USB-Laufwerke ohne Laufwerksbuchstaben heißen dann alle gleich "USB-Massenspeichergerät", es sei den, <u>USBDLM sorgt für aussagekräftige Namen...</u>

Bekommen die Laufwerke immer wieder Laufwerksbuchstaben, ist eine andere Software am Werk, die die Buchstaben bei passender Gelegenheit überprüft und ggf. zuordnet.

Bekannt dafür ist das U3 Launchpad auf U3-Sticks (die U3launch.exe auf dem Pseudo-CDROM), siehe

kein Medium, kein Buchstabe.

Ab Windows Vista gibt es zudem den Effekt, dass pro USB-Wechseldatenträger im Gerätemanager ein (mit Code 10) nicht gestartetes WPD-Gerät erscheint. Da kann USBDLM nichts dafür, es liegt allein am fehlenden Laufwerksbuchstaben, siehe Probleme. Unter Windows 8.1 hat Microsoft das Problem behoben.

LW-Buchstaben abhängig von bestimmten Kriterien

Ab V4.0 kann man die Buchstaben abhängig von einer Vielzahl an Kriterien vergeben.

Man braucht dann mehrere Abschnitte [DriveLetters], in denen jeweils andere Kriterien definiert sind. Ab V5.0 ist keine Nummerierung der Abschnitte mehr nötig, funktioniert aber wie gehabt: Die Abschnitte werden von Nummer 1 an aufwärts bis 99 eingelesen, danach bis zu 100 Abschnitte [DriveLetters] ohne Nummer.

Analog funktioniert das für das Festlegen von Autorun-Ereignissen mittels nummerierter Abschnitte [OnArrival], [OnRemoval], u.a., angepassten Balloontips, u.v.a.

Der erste Abschnitt mit passenden Kriterien wird verwendet (und nur dieser). Das bedeutet, dass man immer vom speziellen Fall aufwärts zu den allgemeineren konfiguriert.

Benutzt man Abschnitts-Nummern dann am besten in 10er-Schritten - dann kann man problemlos mal einen einfügen ohne neu nummerieren zu müssen. Oder ab V5.0 weglassen, dann kann man beliebig umsortieren oder einfügen.

UsbDriveInfo analysiert die USBDLM.INI in dem Ordner in dem USBDLM als Dienst registriert ist, wenn das ein anderer Ordner als der eigene ist, gibt's eine Warnung.

UsbDriveInfo zeigt für jedes Laufwerk welche Abschnitte zuständig sind (nach Ändern der INI per F5-Taste aktualisieren) und fertige Kriterien zum Kopieren und Einfügen in die INI. [DriveLetters]:33 heißt z.B. Abschnitt DriveLetters in Zeile 33.

```
====== Storage Volume on Disk 1 in Partition 1 ========
MountPoint
          = Z·\
Volume Label = King16Back
Volume Size = 15.6 GB / 14.6 GiB / 15.619.129.344 Bytes
Volume Name = \\?\Volume{09247b47-82df-11e4-b297-001d729ad2b2}\
         = STORAGE\VOLUME\ ?? USBSTOR#DISK&VEN KINGSTON&PROD DATATRAVELER 111&REV PMAP#...
DeviceID
DeviceID USB = USB\VID_0951&PID_1693\001CC0C60DBDBCC0C000010B
INI Sections = [DriveLetters]:33, [OnArrival]:38
             - USBDLM Criteria
MinVolumeSize=14G
MaxVolumeSize=16G
MinDriveSize=14G
DeviceID=USBSTOR\DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP\001CC0C60DBDBCC0C000010B&0
DeviceID=USB\VID 0951&PID 1693\001CC0C60DBDBCC0C000010B
DeviceID=Kingston DataTraveler 111 USB Device
PortName=PCI\VEN 8086&DEV 283A&SUBSYS 20AB17AA&REV 03\3&21436425&0&D7-3
VolumeLabel=King16Back
VolumeSerial=A4605DC1
DriveType=REMOVABLE
BusType=USB
PartitionNumber=1
DiskSignature=C3072E18
```

Standard-Kriterien sind:

BusType=USB und DriveType1=REMOVABLE DriveType2=FIXED Das heißt, wenn keine Kriterien angegeben werden, ist ein Abschnitt für alle USB-Sticks und USB-Festplatten zutreffend, was dem Verhalten von USBDLM bis V3.x entspricht. Abweichend von V3.x sind Multislot-Cardreader nicht mehr ausgenommen.

Gibt man einen anderen BusType bzw. DriveType an, wird die jeweilige Vorgabe natürlich außer Kraft gesetzt.

<u>Untypisches</u> Beispiel zum Verdeutlichen der Funktionsweise, Kriterien in grün, Ziel-Buchstaben in violett:

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

Dieser Abschnitt wird benutzt, wenn ein Laufwerk mit der Geräte-ID USB\VID_067B&PID_2517 **oder** USB\VID_1234&PID_5678 angeschlossen wurde. Dem Laufwerk wird dann X: zugeordnet falls frei, sonst Y:. Sind beide Buchstaben belegt, wird der erste freie Buchstabe zugeordnet. Soll in diesem Fall kein Buchstabe zugeordnet werden, konfiguriert man noch Letter3=-

Typisches Beispiel um o.g. Laufwerke auf je einem Buchstaben zu haben:

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

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

Sprich, für jedes Laufwerk dass eine abweichende Behandlung erfahren soll, braucht's einen eigenen Abschnitt.

Weitere Beispiele:

1. FireWire-Laufwerke auf F, USB-Laufwerke ab 10GB auf U, alle anderen USB-Sticks- und Platten auf W oder Z:

;FireWire auf F: [DriveLetters] BusType=FireWire Letter=F

;USB >10GB auf U: [DriveLetters] MinDriveSize=10GB Letter=U ;alle übrigen USB-Laufwerke auf W: oder Z: [DriveLetters] Letters=W, Z

2. USB-Multislot-Cardreader auf L, M, N, O; alle anderen USB-Sticks- und Platten auf W oder Z:

;MultiSlotCardReader auf L:, M:, N: und O: [DriveLetters] DeviceType=MSCR Letters=L,M,N,O

;alle übrigen USB-Laufwerke auf W: oder Z: [DriveLetters] Letters=W,Z

Es können auch mehrere Kriterien gleichen Typs - dann nummeriert - angegeben werden. Davon muss dann eines passen.

Auch mehrere Kriterien verschiedenen Typs sind möglich - dann muss ein Eintrag jeden Typs passen.

Bsp: USB- und FireWire-Laufwerke ab 10GB auf F:

[DriveLetters]
BusTypes=USB, FireWire
MinDriveSize=10GB
Letter=F

Bsp: CF-Karten im PCMCIA-Slot auf P:

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

Sowas funktioniert nicht:

;USB-Festplatten auf K oder L: [DriveLetters] DriveType=fixed Letters=K,L

;eine bestimmte USB-Festplatte auf I:
[DriveLetters]
DriveType=fixed
DeviceID1=USBSTOR\DISK&VEN_WDC_WD20&PROD_00BB-00GUA0
Letters=I

Warum? Der erste Abschnitt "fängt" alle USB-Platten ein, der zweite Abschnitt kommt deshalb nie zum

Zuge.

Deshalb immer vom speziellen Fall aufwärts zu den allgemeineren konfigurieren.

Die verschiedenen Kriterien sind in den folgenden Abschnitten beschrieben, hier eine übersicht:

MinVolumeSize Mindestgröße des VolumesMaxVolumeSize Maximalgröße des Volumes

MinDriveSize Mindestgröße des gesamten Laufwerks
 MaxDriveSize Maximalgröße des gesamten Laufwerks

• DeviceID Geräte-ID des Volumes, des Laufwerks, des Controllers oder der "Friendly

Name"

PortName
 VolumeLabel (*)
 VolumeSerial
 Name des USB-Ports, z.B. 3-1
 Datenträgerbezeichnung
 Seriennummer des Volumes

• DriveType z.B. REMOVABLE, FIXED, CDROM, FLOPPY, RAMDISK

• DeviceType z.B. MSCR, TrueCrypt, BitLocker

• BusType z.B. USB, FireWire, ATA, ATAPI, SCSI

• PartitionNumber Partitionsnummer

• DiskSignature Disk-Signatur

• FileExists (*) Vorhandensein einer Datei

FileNotExists (*)
 Nichtvorhandensein einer Datei
 DirExists (*)
 Vorhandensein eines Verzeichnisses

DirNotExists (*)
 Nichtvorhandensein eines Verzeichnisses
 UserName
 Nutzername des angemeldeten Nutzers

UserName Nutzername des angemeldeten Nutzers
 UserGroup Nutzergruppenname des angemeldeten Nutzers

• UserIsAdmin 1 wenn Nutzer ein Admin ist, sonst 0

• ComputerName Computer-Name, z.B. für einheitliche USBDLM.INI aber unterschiedliches

Verhalten abh. vom Computer

ProcessIsRunning Ein bestimmter Prozess läuft

ProcessIsNotRunning Ein bestimmter Prozess läuft nicht

OsVersion Eine minimale oder maximale Betriebssystem-Version

(*) kann zu Verzögerungen führen oder ganz fehlschlagen, da dafür das Dateisystem fertig eingebunden sein muss; normalerweise problemlos

Hinweis: Frei erfundene Kriterien funktionieren nicht.

Buchstaben abhängig von der Laufwerks-Größe

Die Laufwerks-Größe kann sowohl auf den gesamten Datenträger (die Disk) als auch auf das einzelne Laufwerk (das Volume) bezogen werden.

Die Größe wird sinnvollerweise mit Einheit wie KB, MB oder GB versehen, sonst sind Bytes gemeint.

Die Einheiten verstehen sich 10 hoch n Bytes - ein MB sind also 1.000.000 Bytes

Eine 40GB-Platte hat typischerweise ein Größe von knapp über 40GB, eine einzelne Partition darauf aber meist nur knapp unter 40GB!

Wenn die Grenze z.B. bei 40GB liegen soll, also besser etwas weniger angeben, z.B. 39GB. Nur ganze Zahlen sind zulässig.

Es kann jeweils eine maximale oder eine minimale Größe angegeben werden, verfügbare Einträge sind also: MinVolumeSize

MinDriveSize

MaxVolumeSize

MaxDriveSize

Beispiel:

;USB-Datenträger ab einschließlich 40GB auf U: [DriveLetters] MinDriveSize=39GB Letter=U

;USB-Partition ab einschließlich 10GB auf X: [DriveLetters] MinVolumeSize=9GB Letter=X

Da der Autor immer wieder mal DiskSize statt DriveSize versucht hat, funktioniert das auch...

LW-Buchstaben abhängig von der Geräte-ID

Das Vorgeben von Buchstaben abhängig von der Geräte-ID ist etwas unhandlich zu konfigurieren. Im folgenden wird Geräte-ID "Device-ID" genannt. Genaugenommen ist es die "Instance ID", auf Microsoft-Deutsch "Geräteinstanzkennung".

Beispiel:

[DriveLetters]

DeviceID=USB\VID 067B&PID 2517 DeviceID=USB\VID 1234&PID 5678 Letter1=U

[DriveLetters] DeviceID=USB\VID 9876&PID 5432 Letter=X

Auch der "Friendly Name" - wie von UsbDriveInfo gezeigt - kann benutzt werden. Hier muss ein Leerzeichen enthalten sein, damit USBDLM sieht, dass es ein "Friendly Name" ist. Dieser wird nur bei Bedarf ermittelt.

Es kommt auch vereinzelt vor, dass das Ermitteln des "Friendly Name" fehlschlägt - eine 'richtige' Device-ID ist zuverlässiger und schneller.

[DriveLetters] DeviceID=Corsair Flash Voyager Letter=V

Wenn eine Device-ID konfiguriert ist, sind der Standard BusType=USB und die Standard-Laufwerkstypen REMOVABLE und FIXED außer Kraft gesetzt, da eine Geräte-ID ausreichend eindeutig ist. Bei einem U3-Stick ist die Geräte-ID des USB-Gerätes ("Curl DevID") aber ungeeignet, da diese sowohl für das gefälschte CDROM-Laufwerk als auch für den Wechseldatenträger gilt. Hier besser die "Drive DevID" nutzen.

Die Device-ID muss man sich erstmal ermitteln. Dazu das betreffende Laufwerk anschließen und UsbDriveInfo starten. Als Device-ID kann die des Laufwerks oder des Controllers, sowie der "Friendly Name" genutzt werden.

UsbDriveInfo zeigt dies kopierfertig an, wenn im Menu Options "Show USBDLM criteria" angehakt ist.

MinVolumeSize=14G MaxVolumeSize=16G

MinDriveSize=14G

MaxDriveSize=16G

DeviceID=USBSTOR\DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP\001CC0C060DBDBCC0C000010B&0

DeviceID=USB\VID_0951&PID_1693\001CC0C60DBDBCC0C000010B

DeviceID=Kingston DataTraveler 111 USB Device

PortName=6-3

PortName=PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7-3

VolumeLabel=King16Back VolumeSerial=A4605DC1

DriveType=REMOVABLE DeviceType=DiskReadOK

BusType=USB PartitionNumber=1 DiskSignature=C3072E18

Der String-Vergleich erfolgt unabhängig von Groß-/Kleinschreibung. Ein beliebiger Teil des Strings kann eine Übereinstimmung ergeben. Es müssen mindestens 8 Zeichen konfiguriert werden. Ab V4.2 können alternativ <u>Wildcards</u> benutzt werden.

Für eine große Zahl Device-IDs kann auf eine externe Datei verwiesen werden:

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

In der Text-Datei steht dann je Zeile eine Geräte-ID, ohne Einleitung. Leerzeilen sind erlaubt, Zeilen die mit einem Semikolon beginnen, sind Kommentare. Beispiel:

;Stick vom Admin (der Dativ ist dem Genitiv sein Tod) USB\VID_067B&PID_2517 USB\VID_1234&PID_56789

;Sticks vom Chef USB\VID_9876&PID_5432 USB\VID_4567&PID_4321

;Sticks vom Fußvolk USB\VID_3456&PID_3543 USB\VID_3456&PID_1432

Nutzung der USB-Seriennummer

Im oben genannten Beispiel ist die Device-ID 'USB\VID_067B&PID_2517\6&12115AD4&2&1'. In den letzten Teil (hinter dem letzten Schrägstrich) schreibt Windows die USB-Seriennummer. Dummerweise ist der hier verwendete USB-Stick 'Corsair Flash Voyager 2GB' (altes Modell) einer, bei dem die USB-Seriennummer eingespart wurde. Deshalb wurde dieser Teil der Device-ID von Windows generiert, was am mehrfachen Vorkommen des & Zeichens zu erkennen ist. An jedem USB-Port wird eine neue Nummer generiert.

Soll so ein Gerät unter Verwendung der Geräte-ID als Kriterium an jedem USB-Port den selben Buchstaben bekommen, muss der letzte Teil der Geräte-ID weggelassen werden, also z.B. so: DeviceID1=USB\VID_067B&PID_2517

Bei Geräten, die eine Seriennummer haben, kann man den ganzen String verwenden oder einfach nur die Seriennummer, da sie in der Geräte-ID vorkommt (vorausgesetzt sie ist mindestens 8 Zeichen lang):

[DriveLetters]
DeviceID=1234567890ABCD
Letter=V

Wenn man dann noch Letter1=-- für alle anderen USB-Laufwerke ([DriveLetters]) konfiguriert, hat man den einzigen funktionierenden USB-Stick und ist plötzlich ganz wichtig. USB-Floppys und CD-ROMs sollten dann auch gesperrt werden:

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

Oder kürzer:

[DriveLetters]
DriveType=any
Letter=--

Nicht selten sind die Seriennummern aller Exemplare identisch - ein Verstoß gegen die USB-Massenspeicher-Spezifikation (oder eine sehr eigenwillige Interpretation), siehe auch hier: http://www.uwe-sieber.de/usbtrouble.html

LW-Buchstaben abhängig vom USB-Port

Das Ganze ist etwas unhandlich zu konfigurieren, z.B. so:

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

;für mehrere Ports, z.B. alle an einem Hub der an 5-1 hängt
[DriveLetters]
PortName1=5-1-1
PortName2=5-1-2
PortName3=5-1-3
PortName4=5-1-4
Letters=X
```

Oder kurz unter Nutzung von Wildcards beim PortName:

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

Den Portnamen muss man sich erstmal ermitteln: Dazu an die betreffenden Ports ein USB-Laufwerk anschließen und UsbDriveInfo starten, wo man dann die Port-Namen ablesen kann. UsbDriveInfo gibt für jedes USB-Laufwerk sowas aus:

```
MountPoint
Volume Label
              = King16Back
              = 15.6 GB / 14.5 GiB / 15.619.129.344 Bytes
Volume Serial = A460-5DC1
FileSystem BS = NTFS
Volume Name
               = \\?\Volume{09247b47-82df-11e4-b297-001d729ad2b2}\
              = \Device\HarddiskVolume21
Partition Device = \\.\Harddisk1Partition1
            = STORAGE\VOLUME\_??_USBSTOR#DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP#...
DeviceID Drive = USBSTOR\DISK&VEN_KINGSTON&PROD_DATATRAVELER_111&REV_PMAP\001CC0C60DBDBCC0C000010B&0
DeviceID USB = USB\VID_0951&PID_1693\001CC0C60DBDBCC0C000010B
Device ID HostCtl = PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7
USB Friendly Name = Kingston DataTraveler 111
USB Port Name
             = [DriveLetters]
```

Für den Port, an dem dieses Laufwerk angeschlossen ist, wäre die Einstellung für 'PortName' 3-2. Wenn noch ein USB-Hub dazwischen ist, wäre ein typischer Port-Name 3-2-1, bei zwei Hubs 3-2-1-1 usw. 7-Port-Hubs sind manchmal zwei hintereinander geschaltete 4-Port-Hubs - dann nicht wundern, wenn bei 4 Ports ein Glied mehr als erwartet in der Kennung steht. 10-Port-Hubs können ein 7-Port-Hub plus 4-Port-Hub sein.

Ein-Port-Hubs werden ab V4.4 mitgezählt, z.B. eventuell manche aktive USB-Verlängerung, die ich bisher gesehen habe, waren 4-Port-Hubs bei denen nur ein Port herausgeführt wurde.

USB3-Controller haben separate logische Ports für USB1/2 und 3-Geräte, so dass sich die Portnummer unterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem, ob ein USB1/2- oder 3-Gerät angeschlossen ist, siehe hier: http://www.uwe-nterscheidet, je nachdem ist of the siehe hier: http://www.uwe-nterscheidet, so dass siehe hier: http://www.uwe-nterscheidet, so dass siehe hier ist of the siehe hi

sieber.de/usbtreeview.html#USB3

Das ist aber nur bei Root-Hubs der Fall, bei normalen USB-Hubs gibt es dieses Doppelleben nicht.

Beispiel, um an einem USB3-Port den Buchstaben X: zu bekommen, egal, ob ein USB1/2- oder 3-Gerät angeschlossen wird:

[DriveLetters]
PortName1=3-2
PortName2=3-5
Letters=X

Die erste Nummer im Port-Namen ist die des Root-Hubs. USBDLM nutzt hier die Position des Root-Hubs in der Liste die Windows meldet. Diese Nummer ist leider nicht konstant, bei mehreren Windows-Installationen auf einem Rechner kann es schon Unterschiede geben. Ebenso kann sich die Nummer eines Root-Hubs ändern, wenn ein USB-Host-Controller hinzugefügt oder entfernt wird.

Wenn man dieses Problem hat (z.B. mit einem PCMCIA-USB-Controller), kann man auf die "lange" Notation zurückgreifen. Hier wird die Nummer des Root-Hubs durch die Geräte-ID seines Host-Controllers ersetzt, der Rest bleibt gleich.

Im Beispiel oben war die Device-ID des Host-Controllers PCI\VEN_8086&DEV_283A&SUBSYS_20AB17AA&REV_03\3&21436425&0&D7.

Der sich ergebende lange Portname wäre somit

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

Wenn immer nur einen Hostcontroller mit dieser ID im System gibt, kann man per Wildcard kürzen, etwa so:

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

Nicht-USB-Laufwerke

Ab V4.7 funktioniert das auch mit Laufwerken, die eine "Location" verfügen, sowas wie "Channel 1, Target 0, Lun 0". USBDLM generiert daraus dann eine Kurznotation indem es die Ziffern herauspickt, z.B. 1-0-0. Typischerweise haben ATAPI-, SCSI- und SATA-Laufwerke so eine Location. Alternativ kann auch die "Location" so wie von UsbDriveInfo angezeigt verwendet werden, inklusive Wildcards. Nicht vergessen auch einen <u>BusType</u> zu konfigurieren, da es ja keine USB-Laufwerke sind!

Beispiel für ein ATAPI CD/DVD-Laufwerk auf X:

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

Oder mit der Kurznotation für den Port:

[DriveLetters]

DriveType=CDROM BusType=ATAPI PortName=0-1-0 Letters=X

Buchstaben abhängig vom der USB-Geschwindigkeit

Ab V5.0 kann die maximal USB-Geschwindigkeit des USB-Laufwerkes und die tatsächliche Geschwindigkeit der USB-Verbindung als Kriterium genutzt werden, etwa so:

[DriveLetters] UsbSpeed=High-Speed Letters=H

[DriveLetters] UsbSpeed=SuperSpeed Letters=S

Nutzerbringender ist es bei <u>angepassten BalloonTips</u>:

; USB-SuperSpeed-Laufwerk das mit High-Speed oder weniger verbunden ist:

 $[Balloon Tips On \pmb{Arrival}]$

UsbSpeed=Low-Speed

UsbSpeed=Full-Speed

UsbSpeed=High-Speed

UsbMaxSpeed=SuperSpeed

Title=%FriendlyName% %BusType% %UsbMaxSpeed% Device at %UsbSpeed% - %DriveSize% Text=mounted to %Root% (%FsName%)

Buchstaben abhängig vom Laufwerks-Namen

Der Laufwerks-Name (auch "Datenträger-Bezeichnung", "Volume Label") ist das, was man im Windows Explorer bei Rechts-Klick auf ein Laufwerk -> Eigenschaften im erscheinenden Dialog eingeben kann. Meistens...

Das was der Windows Explorer in Laufwerksliste als Namen anzeigt kann sowohl die Datenträger-Bezeichnung als auch eine Einstellung aus der Registry sein. Das was man beim Einfach-Klick auf ein Laufwerk als Name eingeben kann ist nicht das Volume-Label sondern nur ein Anzeige-Name der in der Registry hinterlegt wird.

Ganz sicher ist man nur auf der Kommandozeile: Start -> Ausführen, CMD. Das Kommando LABEL zeigt und ändert ggf. die Datenträgerbezeichnung, das "volume label".

Beispiel:

[DriveLetters]
VolumeLabel1=Montag
VolumeLabel2=Mittwoch
VolumeLabel3=Freitag
Letter=U

[DriveLetters]
VolumeLabel1=Dienstag
VolumeLabel2=Donnerstag
VolumeLabel3=Samstag
Letter=X

Laufwerke die Montag, Mittwoch oder Freitag heißen, landen so auf U, Dienstag, Donnerstag, Samstag auf X.

Wildcards können benutzt werden:

[DriveLetters] VolumeLabel=Backup? Letter=X

Würde z.B. für Backup1, Backup2 usw. passen.

Mit einem Minus-Zeichen kann man auf eine leere Datenträgerbezeichnung testen:

[DriveLetters] VolumeLabel=-Letter=U

Als Laufwerksbuchstaben nutzen

Ab V4.6 kann USBDLM die Datenträgerbezeichnung direkt als Laufwerksbuchstaben nutzen. Dazu gibt's die neue USBDLM-Variable %LetterFromLabel%.

Hierfür wird ein Buchstabe aus der Datenträgerbezeichnung extrahiert - gesucht wird nach:

- dem letzten Buchstaben dem ein Doppelpunkt folgt und ein Nicht-Buchstabe vorangeht, z.B.
 "Laufwerk X:" oder "X: Laufwerk"
- dem letzten Buchstaben der von Nicht-Buchstaben eingeschlossen ist, z.B. "Laufwerk X", "Laufwerk_X", "Laufwerk (X)", "X-Laufwerk"

Als Nicht-Buchstabe gilt der Beginn, das Ende, das Leerzeichen sowie die Zeichen !"#\$%&'()*+,-./:;<=>? @[\]^_`{|}~~%.

Wenn nichts passendes gefunden wird, ist die Variable %LetterFromLabel% leer.

Hinweis: Der Doppelpunkt ist nur auf NTFS-Laufwerken ein zulässiges Zeichen in der Datenträgerbezeichnung.

Beispiel: Immer versuchen, den Buchstaben aus der Datenträgerbezeichnung zu nutzen, Z: falls keiner da

[DriveLetters] Letter=%LetterFromLabel% Letter=Z

Um Fehlinterpretationen mit fremden Laufwerken zu vermeiden, kann man ein VolumeLabel-Kriterium mit den gewünschten Namensschema konfigurieren, so dass der Abschnitt nur zum Zuge kommt, wenn eine wirklich passende Bezeichnung gefunden wird:

Beispiel: Datenträgerbezeichnung endet auf Unterstrich plus ein Zeichen (z.B. Drive_X):

[DriveLetters] VolumeLabel=*_? Letter=%LetterFromLabel% Letter=Z

Hier noch einige geeignete Wildcards für das VolumeLabel-Kriterium:

- *_? beginnt mit beliebigen (oder keinen) Zeichen, endet auf Unterstrich gefolgt von genau einem Zeichen; Bsp: "Laufwerk X"
- *? beginnt mit beliebigen (oder keinen) Zeichen, endet auf Leerzeichen gefolgt von genau einem Zeichen; Bsp: "Laufwerk X"
- * ?: beginnt mit beliebigen (oder keinen) Zeichen, endet auf Leerzeichen plus ein Zeichen plus Doppelpunkt; Bsp: "Laufwerk X:"

• ?: * beginnt mit genau einem Zeichen plus Doppelpunkt plus Leerzeichen, gefolgt von beliebigen (oder keinen) Zeichen; Bsp: "X: Laufwerk"

Natürlich kann man auch etwas spezifischeres, wie "USB_W", "USB_X", "USB_Y" nutzen, um Überraschungen mit fremden Laufwerken zu vermeiden, die zufällig eine passende Bezeichnung haben:

[DriveLetters] VolumeLabel=USB_? Letter=%LetterFromLabel% Letter=Z

Zum Lesen des Laufwerk-Namens muss das Dateisystem gemountet sein - das kann bei langsam antwortenden Laufwerken zu Verzögerungen führen, oder auch dazu, dass es gar nicht funktioniert, wenn die maximale Wartezeit überschritten wird. Diese lässt sich global im Abschnitt Settings in Millisekunden konfigurieren, Standard sind 10000, also zehn Sekunden:

[Settings] VolumeInformationMaxWait=10000

Buchstaben abhängig vom der Volume-Seriennummer

Die Volume-Seriennummer ist ein 32-Bit Zufallswert, der beim Formatieren eines Laufwerks generiert wird.

Der kann vom Nutzer nicht ohne weiteres geändert werden, ist somit u.U. besser als Kriterium geeignet als die Datenträgerbezeichnung. Außerdem gibt's beim Ermitteln keine Verzögerungen, da USBDLM sie direkt von der Disk liest, und nicht wie die Datenträgerbezeichnung über das Dateisystem des Volumes.

Seriennummer ermitteln:

Start -> Ausführen CMD

C:\>dir u:*.x Datenträger in Laufwerk U: ist TEST Volumeseriennummer: B92B-8000

'*.x' ist eine Dateinamen-Maske, die (meist) null Treffer liefert.

UsbDriveInfo zeigt die Seriennummer auch:

Beispiel:

[DriveLetters] VolumeSerial=B92B-8000 Letter=U

Oder verschiedene Volumes mit verschiedenen Serials (nacheinander) auf einem Buchstaben:

[DriveLetters]

VolumeSerial=B92B-8000 VolumeSerial=34AB-E831 VolumeSerial=A34C-3080 Letter=U

Selbst ändern kann man die Volume-Seriennummer mit dem Kommandozeilen-Tool <u>VolumeID von SysInternals</u>.

So kann sich z.B. eine Reihe Laufwerke mit aufeinanderfolgenden Serials bauen und Wildcards benutzen:

[DriveLetters] VolumeSerial=BABA-00?? Letter=B

Laufwerks-Buchstaben abhängig vom Laufwerks-Typ

Folgende für USBDLM relevanten Laufwerks-Typen (DriveType) sind definiert:

FLOPPY Diskettenlaufwerke

REMOVABLE sonstige Wechseldatenträger, z.B. Kartenleser, aber auch die meisten USB-Sticks

FIXED Festplatten

CDROM CD/DVD-Laufwerke

UNKNOWN unbekannt, manche Kartenleser in Laptops

RAMDISK RAM-Disk (viele RAM-Disks erscheinen als FIXED)

ALL / ANY Beliebiger Typ (Vorsicht!)

Beispiel für USB-CD-ROM auf R:

[DriveLetters]
DriveType=CDROM
Letter=R

Beispiel für ATAPI-CD-ROMs auf L: und M:

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

Wenn kein DriveType angegeben ist, gilt für einen Abschnitt die Vorgabe REMOVABLE oder FIXED, also Sticks, Cardreader, Festplatten und sonstige USB-Wechsellaufwerke wie USB-ZIP, USB-JAZ, USB-MO...

Sollen alle Laufwerks-Typen zutreffend sein, kann kurz "ALL" konfiguriert werden. Um z.B. für alle Nicht-Admins für alle USB-Laufwerke den Buchstaben wegzunehmen konfiguriert man:

[DriveLetters]
UserGroup=Administratoren
Letters=X,Y,Z

;keine USB-Laufwerke für alle anderen [DriveLetters] DriveType=ALL Letter=--

Niemals sowas konfigurieren:

;alle anderen [DriveLetters] BusType=ALL DriveType=ALL Letter1=-

Das würde die Buchstaben **aller** Laufwerke entfernen, ob USB oder nicht ist egal dank BusType=ALL.

Siehe auch <u>Buchstaben abhängig vom aktiven Nutzer</u>.

Laufwerks-Buchstaben abhängig vom Geräte-Typ

Der Geräte-Typ ist etwas USBDLM-spezifisches, um spezielle Geräte gesondert behandeln zu können.

Definiert sind:

- MSCR -> Multislot-Cardreader (ist immer auch CardReader)
- CardReader -> Laufwerke mit charakteristischen Zeichenketten im Name, wie "Card" oder "Reader"
- U3 -> U3-Stick oder eine Festplatte mit virtuellem CDROM, z.B. für Verschlüsselungs-Software
- Floppy -> Disketten-Laufwerk
- VirtualDrive -> Laufwerk, das kein vollwertiges Volume ist, z.B. Subst-, Netz, TrueCrypt-Laufwerke
- SubstDrive -> Subst-Laufwerk
- DokanDrive -> Virtuelles Laufwerk, erzeugt durch Hiroki Asakawas "user-mode file system library"
- TrueCrypt -> Volume dass im ersten Sektor Zufallsdaten enthält, also warscheinlich ein Container für TrueCrypt oder einer anderen Verschlüsselungs-Software ist (z.B. VeraCrypt, FreeOTFE)
- ReadSharingViolation -> Laufwerk beim dem das Lesen mit ERROR_SHARING_VIOLATION fehlschlägt, z.B. ein bereits gemounteter TrueCrypt-Container
- TrueCryptVolume -> ein gemountetes <u>TrueCrypt</u>-Laufwerk
- VeraCryptVolume -> ein gemountetes <u>VeraCrypt</u>-Laufwerk
- OtfeVolume -> ein gemountetes OTFE-Laufwerk
- BitLocker -> ein BitLocker-Volume
- AllZero -> erster Sektor enthält nur Nullen (unformatiert oder Nicht-Windows-Laufwerk)
- ReadError -> Vom Laufwerk konnte nicht gelesen werden
- UnrecognizedVolume -> Windows sagt ERROR_UNRECOGNIZED_VOLUME, z.B. Ext3-Laufwerk
- WindowsRecovery -> Windows Recovery-Partition (Partitions-Typ 0x27 oder PARTITION_MSFT_RECOVERY_GUID bei GPT)

UsbDriveInfo zeigt den oder die zutreffenden Geräte-Typen an.

VirtualDrive, SubstDrive, DokanDrive, TrueCryptVolume, VeraCryptVolume, OtfeVolume betreffen nur virtuelle Laufwerke und sind somit in DriveLetters-Abschnitten nicht sinnvoll.

Beispiel für Multislot-Cardreader auf L, M, N, O

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

Beispiel für Multislot-Cardreader in ein NTFS-Verzeichnis und Nutzung des Geräte-Names:

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

Unter C:\CardReader (der Ordner wird ggf. angelegt) werden dann Ordner anhand der Gerätenamen der Slots angelegt, z.B. 'USB2.0 CF_MD', 'USB2.0 SD_MMC' usw., vorausgesetzt, der Reader hat für seine Slot 'sprechende' Gerätenamen wie die eben genannten. Oft heißen alle Slots gleich, z.B. 'USB USB', 'Generic USB' oder 'General Flash Disk' - dann hängt USBDLM notgedrungen Ziffern an, wenn es der letzte konfigurierte Mountpunkt ist, sonst wird der nächste probiert.

USBDLM erkennt einen Multislot-Cardreader als solchen, wenn er drei oder mehr Slots hat. Außerdem werden USB-Reader mit zwei Slots als Multislot-Cardreader erkannt, wenn sie CardReader erkannt werden, also charakteristische Namen haben.

Nicht-USB-Cardreader, wie sie oft in neueren Laptops anzutreffen sind, werden standardmäßig nicht behandelt. Das lässt sich aber über die Geräte-ID erzwingen.

Beispiel für Laptop-SD-Reader mit BusType SCSI

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

Beispiel zum Verstecken von Laufwerken, die als TrueCrypt-Container genutzt werden:

[DriveLetters]
DeviceType1=TrueCrypt
DeviceType2=ReadSharingViolation
Letter1=-

Die Erkennung von <u>TrueCrypt</u>-Containern funktioniert nur solange TrueCrypt das Volume noch nicht gemountet hat, da TrueCrypt sich exklusiven Zugriff gönnt und USBDLM den Container dann nicht mehr testen kann. Das Container-Laufwerk bekommt dann ersatzweise den DeviceType ReadSharingViolation.

Beispiel zum Verstecken von U3-AutoRun-CD-ROMs:

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

"U3 ist ein Standard für USB-Sticks, der es erlaubt, auf einem USB-Stick alle Einstellungen, Programme und Daten zu speichern, so dass ein Anwender auf Fremd-Rechnern arbeiten kann, ohne auf diesem Spuren davon zu hinterlassen." Dazu soll beim Anschließen des Stick automatisch das 'U3-Launchpad' starten - ein Startmenü, das sich im Systray einnistet. Da Windows XP die Zeile open= in der autorun.inf aber nur auf CD-ROM-Laufwerken standardmäßig ausführt, hat man U3-Sticks kurzerhand ein zusätzliches virtuelles CD-ROM-Laufwerk mit einer autorun.inf und dem U3-Startprogramm verpasst. Das ist das oben genannte 'U3-Autorun-Laufwerk'. Tolle Lösung, die gerechterweise ab Vista nicht mehr ohne Rückfrage funktioniert. Will man davon nichts wissen, konfiguriert man einfach 'kein Laufwerksbuchstabe' als Mountpunkt, wie oben gezeigt. Dann geht dafür kein Autorun mehr und es wird kein Laufwerksbuchstabe verschwendet.

USBDLM sieht alles als U3 was ein CDROM und ein normales Laufwerk auf einem USB-Gerät vereint. So findet sich ein virtuelles CDROM auch auf Laufwerken mit Verschlüsselungs-Funktion oder auf Werbe-Sticks.

Will man das CD-ROM von U3-Sticks loswerden, aber das einer Festplatte behalten, muss man dieses virtuelle CD-ROM in einem Abschnitt mit niedrigerer Nummer abfangen:

[DriveLetters]
DeviceID=DeviceID des CDROM-Laufwerks
DriveType=CDROM
DeviceType=U3
Letter=R

Um einen U3-Stick in einen normalen zu verwandeln, bietet U3 übrigens ein Uninstall-Tool an: http://u3.com/uninstall

Damit kann man auch seine eigene ISO-Datei zum U3-Autorun-Laufwerk machen. Ein grandioses Einfallstor für Schadsoftware auf zum zufällig finden hinterlassenen USB-Sticks...

U3-Sticks gibt's inzwischen kaum noch, mit dem passenden "Factory Tool" kann man aber vielen Sticks einen CDROM-Teil verpassen.

LW-Buchstaben abhängig vom Bus-Typ

Mit Bus-Typ ist die Anschluss-Art des Laufwerks gemeint, wie USB, FireWire, SCSI usw.

Wenn kein BusType konfiguriert ist, gilt der Standard BusType=USB, außer es ist eine <u>Geräte-ID als Kriterium</u> angegeben - die ist (ab V4.3) eindeutig genug.

UsbDriveInfo zeigt auch Nicht-USB-Laufwerke an, wenn im Menü Options der Haken bei "Show all drives" gesetzt ist.

Mögliche BusTypes sind:

USB

FireWire

SCSI

ATAPI

ATA

Remote

SSA

Fibre

RAID

iSCSI

SATA (ab 2003 Server, XP-64 und Vista)

SAS (ab 2003 Server, XP-64 und Vista)

SD (manche interne Laptop-SD/MMC-Cardreader, abhängig von Kartentyp!)

MMC (manche interne Laptop-SD/MMC-Cardreader, abhängig von Kartentyp!)

Virtual (ab Windows 7)

FileBackedVirtual (ab Windows 7, eingebundene VHD-Dateien)

Unknown (manche interne Laptop-Cardreader, virtuelle Laufwerke, Standard-Floppys)

ANY oder ALL (jeder BusType - **Vorsicht, USBDLM könnte interne IDE oder SATA Laufwerke ummounten!**)

SATA-Platten haben bis einschließlich XP nie den BusType=SATA, der war unter XP noch nicht definiert. Dort erscheinen SATA-Platten entweder als ATA oder SCSI, manchmal auch als RAID, das kann auch unter Vista und Windows 7 der Fall sein. Der Treiber des Festplattencontrollers des jeweiligen Laufwerks legt den Bustype fest. UsbDriveInfo zeigt ihn:

```
MountPoint
Volume Label = Backup 18
Volume Size
Volume Serial = 44E0-DE33
Disk Size
             = 200 GB
Volume Name
                = \\?\Volume{49e8598c-b0d0-11de-b41f-00304884ff2a}\
Partition Name = \Device\Harddisk2\Partition1
Bus Type
              = RAID
Drive Type
              = fixed
INI Sections
              = [DriveLetters]:33, [OnArrival]:38
Volume DevID
                = STORAGE\VOLUME\1&30A96598&0&SIGNATURE379DB9F7OFFSET7E0000LENGTH2E933D8200
Drive DevID = $CSI\DISK&VEN_&PROD_$T3200822A$&REV_.3205$&279143B0&0&000000
Ctrl DevID = PCI\VEN_197B&DEV_2360&$UBSYS_2360197B&REV_02\4&F39E1D8&0&00E0
Host Controller = JMicron JMB36X Controller
Friendly Name = ST3200822AS SCSI Disk Device
```

Bus-Typen bitte mit großer Vorsicht konfigurieren. Wer Unsinn konfiguriert bekommt ihn!

Besser <u>immer</u> zusätzlich eine DeviceID konfigurieren - die ist eindeutig. Und bevor man USBDLM damit arbeiten lässt, besser erstmal UsbDriveInfo die dann zuständigen Abschnitte ("INI Sections") anzeigen lassen...

Beispiel: USB- und FireWire-Platten größer 100GB auf X:

[DriveLetters]
BusType=USB
BusType=FireWire
MinDiskSize=100GB
Letter=X

;alternative Notation mit Komma [DriveLetters] BusTypes=USB,FireWire MinDiskSize=100GB Letter=X

Beispiel für ATA-Platte mit DeviceID 'IC35L120AVV207-0' auf X:

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

Beispiel für internen Laptop-Cardreader mit BusType SCSI:

[DriveLetters]
BusType=SCSI
DeviceID=SCSI\DISK&VEN_O2MICRO&PROD_SD
Letter=X

In diesen zwei Beispielen kann ab V4.3 der BusType auch weggelassen werden - da eine (hoffentlich ausreichend eindeutige) DeviceID konfiguriert ist.

Manche interne Laptop-Cardreader haben einen wechselnden BusType, abhängig von der eingelegten Karte SD oder MMC. Da man davon wohl nur einen im System hat, kann man sich die Geräte-ID hier sparen:

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

Beispiel für virtuelle CD-Laufwerke, die von <u>Elby CloneDrive</u> oder von den <u>Daemon-Tools</u> V3 und V4 angelegt werden auf V, W, X, Y:

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

In OnArrival/OnRemoval-Abschnitten kann für TrueCrypt-Laufwerke (DeviceType TrueCryptVolume) das eigentlich nötige BusType=unknown weggelassen werden, wenn im betreffenden Abschnitt BusType konfiguriert ist.

Buchstaben abhängig von der Partitionsnummer

Laufwerke auf Festplatten haben Partitionsnummern - UsbDriveInfo zeigt sie als Teil des Partition Name:

 $\begin{array}{ll} MountPoint & = E: \\ Volume\ Label & = Backup \end{array}$

Volume Size = 18 GB (NTFS, 4 KB clusters, 3.5 GB free)

Volume Serial = 78B8-C37E

Disk Size = 100 GB

Volume Name = \\?\Volume{770f7510-bce2-11df-8471-806d6172696f}\

Partition Name = \Device\Harddisk0\Partition2

Und kopierfertig bei den Kriterien:

Beispiel zum Verstecken von Partition 2 auf einer bestimmten Festplatte:

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

Nicht Letter=-- konfigurieren, da das die gesamte Festplatte abmeldet!

Auf Wechseldatenträgern ist die Partitionsnummer immer 1 oder 0, da Windows ab 2000 hier keine Partitionen unterstützt und immer nur die erste Partition einbindet.

Buchstaben abhängig von der Disk-Signatur

Jedes Laufwerk mit Partitionstabelle hat eine 32 Bit lange Signatur im ersten Sektor. Wenn die Null ist, fragt Windows, ob der Datenträger "initialisiert" werden soll.

Der Wert ist ein Zufallswert, auf diesem basiert die Wiedererkennung von Laufwerken unter Windows.

Ab V4.7.1 kann USBDLM die Signatur aus Kriterium nutzen. Das kann hilfreich sein, wenn sich mehrere Laufwerke sonst nicht unterscheiden.

Der Wert wird als Little-Endian ULONG interpretiert, wenn z.B. auf der Disk bei Offset 1B8h die Bytes 06 D2 42 3D stehen, dann ist der Hex-Wert 3D42D206.

UsbDriveInfo zeigt die Signatur:

```
MountPoint = E:\
Volume Label = Backup
Volume Size = 18 GB (NTFS, 4 KB clusters, 3.5 GB free)
Volume Serial = 78B8-C37E
Disk Size = 100 GB
Disk Signature = 3D42D206 (06,D2,42,3D)
```

Beispiel für obiges Laufwerk auf Z:

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

Die Signatur muss achtstellig oder mit Wildcards notiert werden.

Bei GPT ist die Disk-Signatur meist Null, hier kann die Disk-ID-Guid genutzt werden, z.B. so:

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

UsbTreeView zeigt sie in den Eigenschaften der Disk.

LW-Buchstaben abhängig vom Vorhandensein einer Datei oder eines Ordners

Beispiel: Laufwerksbuchstabe A: wenn auf dem angeschlossenen Laufwerk die Datei \drive_a.id existiert

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

So funktioniert das nur für USB-Laufwerke, wegen des Standardwertes BusType=USB. Soll das auch für andere als USB-Laufwerke funktionieren, muss man nachhelfen:

[DriveLetters] BusTypes=USB,FireWire,SCSI FileExists=%Drive%\drive_a.id Letter=A

Ab V4.2 können für die Datei Wildcards verwendet werden. Die Auswertung erfolgt dabei durch Windows - die Wildcards arbeiten somit abweichend von den <u>USBDLM-internen Wildcards</u>. Es wird nur genau das angegebene Verzeichnis durchsucht, keine Unterverzeichnisse!

Beispiel zum Starten eines Bildbetrachters beim Auffinden mindestens einer JPG-Datei oder des Ordners "DCIM", wie er von allen Digitalkameras angelegt wird und "VideoPlayer" bei AVI und MPG:

[OnArrival]

FileExists=%Drive%*.jpg FileExists=%Drive%\DCIM open="%ProgramFiles%\Bildbetrachter\Bildbetrachter.exe" %Root%

[OnArrival]

FileExists=%Drive%*.avi
FileExists=%Drive%*.mpg
FileExists=%Drive%*.mpeg
open="%ProgramFiles%\VideoPlayer\VideoPlayer.exe" %Root%

Ab V4.3 kann auf Abwesenheit einer Datei (auch Verzeichnis) getestet werden (File**Not**Exists). Beispiel zum Mounten eines TrueCrypt-Volumes auf X: falls X: frei ist:

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

FileExists und FileNotExist beziehen sich auf Dateien und Ordner. Ab V4.3 kann mit DirExists und DirNotExists auch gezielt auf einen Ordner abgefragt werden:

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

Zum Lesen der Datei muss das Dateisystem gemountet sein - das kann bei langsam antwortenden Laufwerken zu Verzögerungen führen, oder auch dazu, dass es gar nicht funktioniert.

Ab V4.6.9.2 kann die maximale Wartezeit in Millisekunden konfiguriert werden. Standard sind 6,5 Sekunden auf CD/DVD-Laufwerken und zwei Sekunden für alle anderen. Beispiel:

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

FileTimeout ist für alle Datei-Kriterien des jeweiligen Abschnitts zuständig.

LW-Buchstaben abhängig vom aktiven Nutzer

Der aktive Nutzer kann über seinen Nutzernamen und/oder seine Nutzer-Gruppe angegeben werden.

Alternativ zum Namen der Nutzer-Gruppe kann ab V4.7.3 auch die Gruppen-SID angegeben werden. Das ist hilfreich bei Domänen-Nutzern die sich mittels zwischengespeicherter Berechtigungen (cached credentials) anmelden weil sie keine Verbindung zum Domain-Server haben. In dieser Situation können die in der Domäne definierten Gruppennamen nicht ermittelt werden, die SID dagegen schon.

Beispiel für Nutzer-Namen:

[DriveLetters] UserName=Vater UserName=Mutter Letter=X Letter=Y

;keine USB-Laufwerke für die Kids [DriveLetters] UserName=KleinKind UserName=GrossKind DriveType=ANY Letter=-

DriveType=ANY ist nötig, damit auch USB-Floppy- und USB-CDROM-Laufwerke keinen Buchstaben bekommen.

Beispiel für Gruppen-Namen:

[DriveLetters]
UserGroup=Administratoren
Letters=X,Y

;keine USB-Laufwerke für alle anderen [DriveLetters] DriveType=ANY Letter=-

Oder umgekehrt:

;keine USB-Laufwerke für Entrechtete

[DriveLetters]
UserGroup=Benutzer
UserGroup=Gäste
DriveType=ANY
Letter=-

;alle anderen haben's gut [DriveLetters] Letters=X,Y

Oder über "Admin oder nicht":

[DriveLetters] UserIsAdmin=1 Letters=X,Y

;keine USB-Laufwerke fürs Fußvolk [DriveLetters] UserIsAdmin=0 DeviceType=ANY Letter=-

Damit ein bestimmter Stick auch bei einem Rechtlosen funktioniert, kann man <u>davor</u> noch einen Abschnitt mit der Geräte-ID des Sticks einfügen, siehe <u>Buchstaben abhängig von der Geräte-ID</u>.

;der geht immer [DriveLetters] DeviceID=USB\VID_067B&PID_2517 Letters=X,Y

Wenn kein Nutzer angemeldet ist, sind Nutzername und Gruppe leer, es wird also kein Abschnitt benutzt in dem UserGroup oder UserName definiert sind. UserIsAdmin ist 0 wenn kein Nutzer angemeldet ist.

Um einen Abschnitt bei "kein Nutzer angemeldet" wirksam zu machen, kann UserName=- oder UserGroup=- konfiguriert werden.

Wenn Windows nach "Benutzer wechseln" den Anmeldebildschirm zeigt, gilt der letzte aktive Nutzer noch als angemeldet.

LW-Buchstaben abhängig vom Computernamen

Sollen mehrere Computer eine einheitliche Konfiguration haben aber dennoch unterschiedlich reagieren, kann der Computername als Kriterium genutzt werden.

; USB-Laufwerke für die Lehrer [DriveLetters] ComputerName=LehrerRechner ComputerName=OberLehrerRechner Letters=X,Y

;keine USB-Laufwerke für alle anderen [DriveLetters] DriveType=ANY Letter=-

DriveType=ANY ist nötig, damit auch USB-Floppy- und USB-CDROM-Laufwerke keinen Buchstaben bekommen.

LW-Buchstaben abhängig vom einem Prozess

Wozu auch immer, wohl eher in OnArrival-Abschnitten sinnvoll.

; USB-Laufwerke auf U: falls der Taschenrechner läuft [DriveLetters] ProcessIsRunning=CALC.EXE Letters=U

; USB-Laufwerke auf V: falls der Windows-Editor nicht läuft [DriveLetters] ProcessIs**Not**Running=NOTEPAD.EXE Letters=V

; sonst USB-Laufwerke auf Z: [DriveLetters] Letter=Z

Prozessnamen funktionieren nur ohne Pfadangabe, so wie sie im Windows Taskmanager im Reiter "Prozesse" stehen.

Buchstaben abhängig vom Betriebssystem

Wenn man mehrere Computer mit einer einheitlichen USBDLM-Konfiguration betreiben will, aber abhängig vom Betriebssystem doch Unterschiede braucht, kann man die Kriterien MinOsVer und MaxOsVer nutzen.

Beispiel: Laufwerksbuchstaben von mit BitLocker To Go verschlüsselten Laufwerken entfernen und eine Warnmeldung zeigen, wenn das Betriebssystem Vista oder älter ist, also BitLocker To Go nicht unterstützt:

[Driveletters]
MaxOsVer=6.0
DeviceType=BitLocker
Letter=-

[OnArrival]

MaxOsVer=6.0

DeviceType=BitLocker

open="%usbdlmpath%\usbdlm.exe" -balloon -time=20000 -title="%FriendlyName%" -text1="BitLocker verschlüsseltes Laufwerk kann nicht gelesen werden!" -icon=131

Die internen Windows-Versionsnummern sind:

• Windows 2000: 5.0

• Windows XP: 5.1

• Windows XP x64: 5.2

• Windows Server 2003: 5.2

• Windows Vista: 6.0

• Windows Server 2008: 6.0

• Windows 7: 6.1

Windows Server 2008 R2: 6.1

• Windows 8: 6.2

• Windows Server 2012: 6.2

• Windows 8.1: 6.3

• Windows Server 2012 R2: 6.3

• Windows 10: 10.0

Oder einfach UsbDriveInfo starten, das zeigt auch die Versionsnummer des Betriebssystems.

Buchstaben abhängig von der Zeit

1. Tageszeit

Beispiel:

; USB-Laufwerke von 8 bis 12 und 13 bis 18 Uhr auf U, sonst kein Buchstabe

[DriveLetters]

MinDaytime1=08:00

MaxDaytime1=12:00

MinDaytime2=13:00

MaxDaytime2=18:00

Letters=U

[DriveLetters]

Letters=-

Das Ganze funktioniert über einen simplen Textvergleich, also unbedingt z.B. 08:00 schreiben, nicht 8:00!

2. Wochentag

Beispiel:

; USB-Laufwerke von Montag bis Freitag auf U, sonst kein Buchstabe

[DriveLetters]

Weekdays=MO,TU,WE,TH,FR

Letters=U

[DriveLetters]

Letters=-

Die Wochentage werden mit englischen Zwei-Buchstaben-Kürzeln notiert, also MO, TU, WE, TH, FR, SA, SU. Bereiche wie MO-FR werden nicht unterstützt.

3. Vergangene Zeit seit einem Systemereignis

Unterstützte Ereignisse sind:

Boot

UserLogon

UserLogoff

UserLock

UserUnlock

UserDisconnect
UserConnect
UserActivated
UserDeactivated
SleepRequest
Sleep
Resume
Dock
Undock
UndockRequest
UndockCanceled

Notiert jeweils nach einem MinTimeSince oder MaxTimeSince, also z.B. MaxTimeSinceBoot und mit der Zeit in Millisekunden.

Ist ein Ereignis noch nicht aufgetreten, sind Min- und MaxTimeSinceXxx unabhängig von der angegebenen Zeit nicht zutreffend. Außer bei "Boot" ist das bei UsbDriveInfo immer der Fall, da es von den Ereignissen nichts weiß.

Beispiel: Beim Start des USBDLM-Dienstes das Ramdisk-Laufwerk R: ins NTFS-Dateisystem konvertieren, aber nur wenn seit dem Systemstart weniger als eine Minute vergangen ist (anderenfalls wurde USBDLM wohl manuell gestartet, der Versuch es nochmal zu konvertieren wäre unsinnig):

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

Beispiel: Taschenrechner starten bei Resume, aber nur wenn Sleep mindestens eine Minute her ist (anderenfalls ist der Computer wohl unbeabsichtigt gleich wieder aufgewacht):

[OnResume] MinTimeSinceSleep=60000 open=calc

Sonstige Kriterien

Docking-Status

; USB-Laufwerke auf U wenn gedockt, sonst auf X [DriveLetters]
IsDocked=1
Letters=U

[DriveLetters] IsDocked=0 Letters=X

"Sleeping" Status

Während ein Computer schläft gibt's natürlich keine Ereignisse. Diese Kriterium ist für eine spezielle Situation gemacht:

Wenn XP schlafen geht, wird der Nutzer nicht gesperrt, das passiert erst wenn der Computer wieder aufwacht. Die Benachrichtigung fürs Aufwachen kommt nach der für Sperren, somit gibt es einen kurzen Moment wo IsSleeping=1 ist.

Manchmal kommen die Benachrichtigungen doch andersrum, dafür ist das MinTimeSinceResume=10000:

; Bildschirm ausschalten wenn Computer manuell gesperrt [OnUserLock]
IsSleeping=0
MinTimeSinceResume=10000
open="C:\Tools\nircmd.exe" monitor off

NirCmd ist Nir Sofers Kommandozeilen-Tool mit dem viele Windows-Funktionen ohne Nutzerinteraktion ausgelöst werden können:

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

Prioritäten

Bis Version 3.x waren die Prioritäten der Konfiguration fest verdrahtet. Ab Version 4 hat man völlig freie Hand.

USBDLM V4 hat von [DriveLetters1] bis [DriveLetters99] und zuletzt noch [DriveLetters] ausgewertet. Der erste Abschnitt bei dem die Kriterien zum Laufwerk passen (und nur dieser) wird benutzt. Also immer vom speziellen Fall zum allgemeinen hin aufwärts konfigurieren.

Ab USBDLM V5 braucht man in der INI nicht mehr nummerieren, man kann bis zu 100 Abschnitte eines Typs haben, bei nicht nummerierten Abschnitten geht es allein nach der Reihenfolge.

Wenn alle Abschnitte [DriveLetters] keine Neuzuordnung des Laufwerks-Buchstaben ergeben haben, wird noch geprüft, ob der vergebene Laufwerksbuchstabe vom derzeit angemeldeten Nutzer schon als Netz- oder Subst-Laufwerk verwendet wird. Dann ordnet USBDLM den nächsten wirklich freien Buchstaben zu.

Zuletzt wird noch geprüft, ob das neue Laufwerk überhaupt einen Mountpunkt bekommen hat. Falls nicht, wird es dem ersten freien Buchstaben zugeordnet. Natürlich nur wenn das Laufwerk nicht auf "kein Buchstabe" konfiguriert wurde...

AutoRun

Windows AutoRun

Das Windows-AutoRun ist nicht jedermanns Sache: Dumme Fragen, sinnfreies Herumsuchen, Tendenz zu Eigenleben, Vergesslichkeit und Nichtfunktionieren...

Also am besten abschalten und USBDLM überlassen. Abschalten geht am besten mit meinem Tool <u>AutoRunSettings</u>.

Mit Microsoft TweakUI kann man es nicht für Festplatten ausschalten und es aktiviert nebenbei AutoRun für Netz-Laufwerke...

USBDLMs AutoRun-Funktionen

USBDLM bietet zwei Mechanismen zum automatischen Starten von Programmen beim Anschließen eines Laufwerks und beim Einlegen eines Mediums (Karte, Disk, CD/DVD). Das sind zum einen das Ausführen der Zeile open= in einer autorun.inf auf dem jeweiligen Laufwerk und zum anderen AutoRun-Ereignisse, die in der USBDLM.INI konfiguriert werden.

Die jeweilige Kommandozeile wird standardmäßig im Kontext des angemeldeten Nutzers ausgeführt. Wenn kein Nutzer angemeldet ist, wird das Kommando nicht ausgeführt, außer es ist für die betreffende open-Zeile "system=1" gesetzt.

Remote-User werden ab V4.4.3 zwar berücksichtigt, aber immer nur einer. Die Auswahl erfolgt abhängig von der Einstellung "RemoteSessions" im Abschnitt [Settings]:

RemoteSessions=0 Remote-Sessions werden ignoriert

RemoteSessions=1 Remote-Sessions werden berücksichtigt, falls kein lokaler Nutzer aktiv (Standard)

RemoteSessions=2 Lokale und Remote-Nutzer sind gleichberechtigt

Falls mehrere Nutzer gleichwertig angemeldet sind, "gewinnt" der zuletzt angemeldete.

USBDLMs AutoRun funktioniert auch, wenn das Laufwerk in einen NTFS-Ordner gemountet wurde.

Durch Halten der Umschalt-Taste ("Shift") wird das Starten übergangen.

Ab V4.8.8 wird auch das Starten von Dokumenten unterstützt.

open=%drive%\test.txt

würde z.B. den für den aktiven Nutzer eingestellten Texteditor starten.

Wenn der Pfad Leerzeichen enthält, muss dieser in Anführungszeichen eingeschlossen werden, sonst muss USBDLM raten, wo der Pfad endet und die Parameter beginnen. Beispiel:

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

Das ist auch der Fall, wenn Umgebungsvariablen zu Leerzeichen führen:

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

Mit SafeCommandLines=1 im Abschnitt [Settings] ist das immer dann Pflicht wenn die Kommandozeile ein Leerzeichen enthält, auch wenn's nur für den Parameter ist: Beispiel:

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

Scheint offensichtlich, könnte aber auch "C:\Tools\test.exe -t.exe" meinen. Deshalb besser immer den Pfad in Anführungszeichen einschließen:

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

Erstmals konfigurierte Autorun-Ereignisse beim Einlegen und Entnehmen eines Mediums aus einem Kartenleser funktionieren u.U. erst nach einem erneuten Anschließen des Gerätes oder nach einem Neustart des USBDLM-Dienstes.

Das "Arbeitsverzeichnis" ist standardmäßig das Stammverzeichnis des betreffenden Laufwerks. Ein abweichendes kann mit dem Parameter workdir angegeben werden. Das ist z.B. erforderlich, wenn ein Laufwerk "zum sicheren Entfernen" vorbereitet werden soll, da es auf das Arbeitsverzeichnis immer ein offenes Handle gibt.

Beispiel: Beim Anschließen eines USB-Laufwerks mit der Datenträgerbezeichnung "Backup" C:\BatchFiles\make_backup.cmd ausführen und das Laufwerk als Parameter übergeben. Maximal 60 Sekunden warten bis das beendet ist und dann "Hardware sicher entfernen" mittels RemoveDrive, wobei das Arbeitsverzeichnis wie oben beschrieben gesetzt wird:

[OnArrival]

VolumeLabel=Backup

open="C:\BatchFiles\make_backup.cmd" %drive% wait=60s

open="C:\Tools\RemoveDrive.exe" %drive% -L workdir=%windir%

Ohne workdir=%windir% wäre das Stammverzeichnis des Laufwerks das Arbeitsverzeichnis und das würde das "Sichere Entfernen" verhindern.

Ausführbare Dateien werden ab V4.6 zuerst im Windows-Suchpfad (Umgebungsvariable PATH) gesucht, dann im Arbeitsverzeichnis. Diese Änderung erfolgte, da Nutzer oft (wie auch früher in dieser Hilfe vorgemacht) z.B. open=explorer %drive% konfigurieren. So müsste man nur eine Schadsoftware "explorer.exe" nennen, auf einen Stick tun und an einen Rechner mit USBDLM anschließen wo das so konfiguriert ist.

Ein zusätzlicher Suchpfad kann konfiguriert werden, z.B.:

[Settings]

OpenSearchPath=C:\BatchFiles

Die open-Zeile kann auch aus einer autorun.inf extrahiert werden (ein AutoRunKey wird dann aber nicht überprüft):

[OnArrival] open=%drive%\autorun.inf

Die <u>USBDLM-Variablen</u> können dem ausgeführten Programm als Umgebungsvariablen zur Verfügung gestellt werden. Anzugeben ist eine Liste mit den gewünschten Variablen. Beispiel für das Laufwerk (z.B. U:) in der Variable drive

[Settings]

UsbdlmVariablesToOpenEnvironment=%drive%

Beide unten beschriebenen Mechanismen können für beim Nutzer-Login vorhandene Laufwerke ausgeführt werden, als ob sie gerade angeschlossen wurden.

[Settings]
AutoRunOnLogon=1

AutoRun beim Anschießen

1. autorun.inf auf dem angeschlossenen Laufwerk

Die autorun.inf ist ein Windows-Mechanismus. Unter anderem kann über eine Zeile open= Windows beim Anschließen eines Laufwerks oder Einlegen eines Mediums (CD, DVD, Flash-Karte) ein beliebiges Programm ausführen. Mit jeder Windows-Version wurde das weiter eingeschränkt, XP macht es nur noch auf CD/DVD-Laufwerken ohne Nachfrage, was dann zu so tollen Lösungen wie U3-Sticks und USB-Geräten mit Selfinstallern geführt hat, die ihren Autostart über ein virtuelles CD-ROM bewerkstelligen. Gerechterweise geht ab Vista auch das nicht mehr ohne Nachfrage.

USBDLM kann die Zeile open= in einer autorun.inf auf Wechseldatenträgern, Festplatten, CD-ROMs und Floppys ausführen, bei Wechseldatenträgern auch beim Einlegen einer Karte, sonst nur beim Anschließen eines Laufwerks.

Bsp: Nur für Wechseldatenträger:

[Settings]
AutoRunInf=1

Werte für andere Typen, bei Bedarf addieren:

Wechseldatenträger: 1
Festplatten: 2
CDROMs: 4
Floppys: 8
Unbekannt: 16
Remote (Netzlaufwerk): 32

Bsp: Nur für Wechseldatenträger, Festplatten und CD-ROMs:

[Settings]
AutoRunInf=7

Floppys unterstützen keine 'Benachrichtigung beim Einlegen' (insert notification) - daher kann das bei Floppys nur funktionieren, wenn ein externes Laufwerk mit eingelegtem Medium angeschlossen wird.

Beispiel zum Öffnen eines maximierten Explorer-Fensters über eine autorun.inf auf dem angeschlossenen Laufwerk bzw. eingelegten Medium (Der Punkt steht für das aktuelle Verzeichnis, also das Stammverzeichnis des angeschlossenen Laufwerks):

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

Auf x64-Systemen wird zuerst aus dem Abschnitt [Autorun.Amd64] gelesen. Wird ein Wert dort nicht gefunden, wird [Autorun] gelesen.

Dem zu startenden Programm kann über die Zeile openstyle= ein Fenster-Stil vorgeschlagen werden - das ist eine Funktion von USBDLM und wird vom Windows-AutoRun nicht unterstützt. Viele Programme halten sich dran, z.B. der Windows-Editor "notepad.exe", andere nicht, z.B. der Windows-Rechner "calc.exe". Verfügbare Stile sind:

max maximiert
min minimiert
hidden versteckt
noactivate normal, aber Fenster wird nicht aktiviert, bekommt also nicht den Fokus

Da der Autor es selbst versehentlich immer wieder mit windowstyle= versucht hat, funktioniert das jetzt auch...

Ab Vista ist es recht knifflig, das Fenster eines aus einem Dienst heraus gestarteten Programms aktiviert zu bekommen. Falls das gar nicht nötig ist, bitte openstyle=noactivate konfigurieren.

Die autorun.inf bietet natürlich ein hervorragendes Einfallstor für Schadsoftware auf Fremd-Sticks. Deshalb kann USBDLM das Ganze über einen Schlüssel absichern:

[Settings]

AutoRunKey=MeinTotalGeheimerSchlüssel

Nur wenn sich in der autorun.inf die gleiche Zeile befindet (dort im Abschnitt [autorun]), führt USBDLM die open= Zeile aus.

Eine weitere Möglichkeit, etwas für die Sicherheit zu tun, ist, AutoRun ohne Admin-Rechte auszuführen (falls man nicht ohnehin als "Eingeschränkter Nutzer" oder mit aktiver UAC arbeitet).

[Settings]

AutorunInfRestricted=1

Wunder darf man davon nicht erwarten - auch ohne Admin-Rechte kann man Tastatureingaben aufzeichen, Nutzerdateien lesen, schreiben und löschen; nach Hause telefonieren geht auch.

Ab V4.3 funktionieren hier auch die meisten anderen Erweiterungen, wie open1 bis open9, wait, delay usw., siehe unten.

Label und Icon aus der autorun.inf

Ab V4.4 kann USBDLM die Einträge label und icon aus der autorun.inf lesen und in die Registry-Werte DefaultLabel und DefaultIcon für dieses Laufwerk eintragen. Das kann sinnvoll sein wenn die autorun.inf komplett deaktiviert wurde oder einfach nicht funktioniert..

Beides muss separat aktiviert werden, analog zu AutoRunInf, siehe oben.

Beispiel für Label und Icon von CDROM-Laufwerken:

[Settings]
AutoRunInfLabel=4
AutoRunInfIcon=4

2. AutoRun-Ereignisse in der USBDLM.INI - abhängig von Laufwerks-Eigenschaften

[OnArrival] open=calc.exe

Das würde z.B. bei jedem Anschließen eines USB-Laufwerks oder Einlegen einer Flash-Karte den Windows Taschenrechner ausführen.

Es gelten die selben Kriterien wie bei den DriveLetters-Abschnitten, ein Abschnitt ist also standardmäßig nur für USB-Sticks und Festplatten gültig. Für andere Laufwerkstypen braucht's entsprechende Kriterien wie BusType und/oder DriveType. Hier darf man dann auch gefahrlos mit Einstellungen wie BusType=ANY und DriveType=ANY arbeiten.

; für CD- und Netzlaufwerke [OnArrival] DriveType1=CDROM DriveType2=REMOTE open=calc.exe

Das Kommando kann auch ohne Admin-Rechte ausgeführt werden (wenn der aktive Nutzer kein Admin ist, ergibt sich kein Unterschied):

[OnArrival] open=calc.exe restricted=1

oder mit vollen System-Rechten im Kontext "LocalSystem":

[OnArrival] open=calc.exe system=1

Das Fenster der Anwendung wird man aber nie zu sehen bekommen, da man aus "LocalSystem" heraus nicht einfach so auf den Desktop des Nutzers kommt.

Nützlich ist das zum Ausführen von Tools die Admin-Rechte benötigen, der aktive Nutzer aber keine hat, wie unten im Beispiel 6.

Außerdem funktioniert das auch wenn kein Nutzer angemeldet ist. Was man so ausführt sollte sich aber unbedingt selbst beenden, da der Nutzer nichts davon sieht und nicht eingreifen kann.

Mit Windows Vista wurde die "User Account Control", kurz UAC, eingeführt. Das Nutzer-Konto ist dann

geteilt in eines mit eingeschränkten Rechten und eines mit vollen Administrator-Rechten, das was man per "Als Administrator ausführen" nutzt. Es hat den Beinamen "elevated".

Standardmäßig startet USBDLM Prozesse mit dem eingeschränkten Konto. Will man "elevated" starten:

[OnArrival] open=calc.exe elevated=1

Als Parameter für das auszuführende Programm kann man <u>Variablen</u> wie z.B. %drive% das angeschlossene Laufwerk oder %root% dessen Stammverzeichnis übergeben.

Das Ganze geht auch abhängig von Kriterien, genau wie bei der Konfiguration der Laufwerksbuchstaben. Geprüft werden [OnArrival1] bis [OnArrival99], anschließend noch 100 [OnArrival] (ohne Nummer). Der erste Abschnitt mit passenden Kriterien wird benutzt.

Ein zusätzliches Kriterium ist der Laufwerksbuchstabe, Letter=X ist hier also ein Kriterium! Buchstabenlisten funktionieren hier nicht. Ein NTFS-Mountpunkt muß hier mit einem Backslash enden. Außerdem gilt das Standard-Kriterium MinDiskSize=1, so dass OnArrival-Ereignisse nur ausgeführt werden, wenn das Laufwerk mindestens 1 Byte groß ist, sprich, ein Medium eingelegt ist. Um OnArrival auszuführen wenn kein Medium eingelegt ist (wozu auch immer), kann MinDiskSize=- konfiguriert werden.

Dem zu startenden Programm kann über die Zeile openstyle= noch ein Fenster-Stil vorgeschlagen werden, siehe oben.

Bei Bedarf kann noch eine Verzögerung angegeben werden, Standardmäßig in Millisekunden, ab V4.8.9.5 können Einheiten angegeben werden, 's' für Sekunden, 'm' für Minuten und 'h' für Stunden. Maximum sind 24h:

delay=10s

Wenn das Ausführen übersprungen werden soll, kann die Umschalt-Taste ("Shift") gedrückt gehalten werden. Ab Vista funktioniert das nur, wenn im Abschnitt Force=0 gesetzt ist, da der Tastenstatus hier nur mit etwas Aufwand zu ermitteln ist.

Um die Integrität der auszuführenden Datei sicherzustellen, kann ein MD5-Hashwert angegeben werden:

[OnArrival] open=calc.exe md5=DDCD9FCD B7E1956E E69F8E58 B8C8BF0D

Groß- / Kleinschreibung ist egal, ebenso die Leerzeichen. Nur wenn der MD5-Hash der EXE mit dem konfigurierten Wert übereinstimmt, wird das Programm ausgeführt.

Tool zum Ermitteln des MD5-Hash einer Datei: MD5-File.zip

Soll verhindert werden, dass ein Programm ein weiteres Mal gestartet wird, konfiguriert man OneInstance=1:

[OnArrival] open=calc.exe OneInstance=1

Beispiele

Beispiel 1:

- FotoSoftware für Laufwerke mit der Datenträgerbezeichnung "CANON_DC" oder "NIKON_DC", als Parameter die <u>USBDLM-Variable</u> %root%, also das Stammverzeichnis des neuen Laufwerks
- nix für Laufwerk X
- sonst ein maximiertes Explorer-Fenster mit dem Laufwerk

[OnArrival]

VolumeLabel=CANON_DC VolumeLabel=NIKON_DC open="C:\Programme\Foto Software\FotoSoftware.exe" %root%

[OnArrival]

Letter=X

open=

[OnArrival]

open="%windir%\explorer.exe" %root% openstyle=max

Beispiel 2:

- Falls vorhanden, die Datei DATEN.TXT vom angeschlossenen Laufwerk nach C:\Daten kopieren, vorher zwei Sekunden warten:

[OnArrival]

FileExists=%drive%\DATEN.TXT

delay=2000

open="%windir%\System32\cmd.exe" /c copy "%drive%\DATEN.TXT" "C:\Daten"

cmd ist der Windows-Kommandozeilen-Interpreter, /c heißt "Kommando ausführen und dann beenden", copy ist ein Kommando das cmd bekannt ist und Dateien kopiert.

Das Selbe versteckt:

[OnArrival]

FileExists=%drive%\DATEN.TXT open="%windir%\System32\cmd.exe" /c copy "%drive%\DATEN.TXT" "C:\Daten" openstyle=hidden

Beispiel 3:

- Eine TrueCrypt Container-Datei mit Namen geheim.tc automatisch auf T: mounten:

[OnArrival]

FileExists=%drive%\geheim.tc open="C:\Programme\TrueCrypt\TrueCrypt.exe" /v "%drive%\geheim.tc" /L T /q

Beispiel 4:

- Eine als TrueCrypt-Container eingerichtete Partition automatisch auf J: mounten, deren überflüssigen Laufwerksbuchstaben entfernen und das gemountete TrueCrypt-Laufwerk im Explorer öffnen:

;Laufwerksbuchstaben entfernen, ist eh nutzlos [DriveLetters]
DeviceType1=TrueCrypt
DeviceType2=ReadSharingViolation
Letter=-

[OnArrival]

DeviceType=TrueCrypt
open="C:\Programme\TrueCrypt\TrueCrypt.exe" /v %PartitionName% /L J /q

[OnArrival]

DeviceType=TrueCryptVolume open="%windir%\explorer.exe" %root%

%PartitionName% ist eine <u>USBDLM-Variable</u> und wird durch sowas wie \Device\Harddisk2\Partition1 ersetzt, so wie TrueCrypt das erwartet.

Der DeviceType ReadSharingViolation trifft für Laufwerke zu, die bereits durch TrueCrypt gemountet sind. Davon kann USBDLM keine Testdaten lesen - der Versuch wird mit ERROR_SHARING_VIOLATION zurückgewiesen, daher der Name. Dazu kommt es z.B. wenn der USBDLM-Dienst neu gestartet wird. Der Buchstabe soll dann ja weg, nochmal mounten wäre aber falsch. Genau das wird durch obige Konfiguration erreicht.

Hinweis: USBDLM reagiert nur auf "Volumes", das was für gewöhnlich einen Laufwerksbuchstaben hat. Wechseldatenträger haben unter Windows 2000 und höher immer eine Partition (bei Superfloppys gilt der ganze Datenträger als solche) und somit auch immer ein Volume, selbst wenn sie nicht formatiert sind. Deshalb funktioniert USBDLMs AutoRun mit TrueCrypt auch dann wenn ein Wechseldatenträger komplett als TrueCrypt-Container genutzt wird.

Bei Festplatten geht das nicht - hier muss man eine Partition anlegen - die kann dann als TrueCrypt-Container genutzt werden. Eine Festplatte komplett, also ohne Partition, als TrueCrypt-Container zu nutzen ist keine gute Idee, da im ersten Sektor eine Partitionstabelle erwartet wird. Wenn dann stattdessen Zufallsdaten vorgefunden werden, ist das Ergebnis nicht vorherzusagen - Windows kommt dann beim Öffnen der Datenträgerverwaltung mit der Empfehlung, die Platte zu initialisieren, was dem TrueCrypt-Container nicht gut tut...

Ab V4.8 kann USBDLM AutoRuns für ganze Laufwerke ausführen.

Beispiel 5:

Einen USB-Stick auf U: mounten und als Stick_U freigeben, beim "sicheren Entfernen" die Freigabe wieder löschen:

[DriveLetters] Letter1=U

[OnArrival]

Letter=U

open="%windir%\System32\net.exe" share Stick_U=U:\ openstyle=hidden

; beim "sicher entfernen" muss die Freigabe weg, sonst wird's nichts, falls jemand drauf zugreift [OnRemovalRequest]

Letter=U

open="%windir%\System32\net.exe" share Stick_U /D openstyle=hidden

; falls das Laufwerk einfach abgezogen wurde
[OnRemoval]
Letter=U
open="%windir%\System32\net.exe" share Stick_U /D
openstyle=hidden

Mit openstyle=hidden kann man das aufpoppende Konsolenfenster unterdrücken.

Mehrere Open-Kommandos

Ab V4.3 können bis zu 10 Open-Kommandos ausgeführt werden - open1 bis open9 und open. Zusätzliche Parameter wie openstyle, restricted usw. müssen dann jeweils die selbe Nummer tragen. Ab V5 kann man auf die Nummerierung verzichten, eine Zeile open= beginnt einen neuen Parameterblock. "delay" definiert eine Wartezeit bevor die open-Zeile ausgeführt wird. "wait" bewirkt, dass gewartet wird

"delay" definiert eine Wartezeit bevor die open-Zeile ausgeführt wird. "wait" bewirkt, dass gewartet wird bis das Programm das die open-Zeile gestartet hat beendet wurde - erst dann wie die nächste open-Zeile in Angriff genommen.

Beispiel:

- Zweimal den Rechner starten, einmal eingeschränkt, einmal normal:

[OnArrival] open1=calc.exe restricted1=1 open2=calc.exe restricted2=0

Ab V5 braucht man keine Nummerierung mehr, eine open-Zeile beginnt einen Parametersatz:

- Rechner starten, zwei Sekunden später Notepad maximiert starten

[OnArrival]

```
open=calc.exe
delay=2000
```

open=notepad.exe windowstyle=max

- Rechner starten, Notepad sobald Rechner beendet oder 10 Sekunden abgelaufen

[OnArrival] open=calc.exe wait=10s

open=notepad.exe windowstyle=max

Weitere realistische Beispiele:

Beispiel 6:

Einen USB-Stick auf A: mounten, falls A: belegt, dieses Laufwerk aus dem Weg räumen, indem es auf B: umgemountet wird:

;Auf A: mounten falls verfügbar, sonst unmounten [DriveLetters]
Letter=A
Letter=-

;falls ungemountet (Kriterium "kein Mountpunkt"), A: auf B: ummounten und dann A: zuordnen [OnArrival]

Letter=-

open="C:\Tools\ReMount.exe" A: B:

system=1

wait=1000

open="%windir%\System32\MountVol.exe" A: %VolumeName% system=1

%VolumeName% ist eine <u>USBDLM-Variable</u> und wird durch sowas wie

\\?\Volume{aa6d706a-15da-11dc-a38f-0013d31dd4c5}\ ersetzt, so wie das Windows Kommandozeilen-Tool MountVol das erwartet.

ReMount ist mein Kommandozeilen-Tool zum Ändern von Laufwerksbuchstaben. Download: http://www.uwe-sieber.de/drivetools.html#remount

Wait bewirkt, dass das nächsten open erst ausgeführt wird, wenn das aktuell open fertig ist (oder 1000 ms abgelaufen sind, was bei ReMount aber nicht passieren wird).

Durch system=1 wird das jeweilige Kommando im Kontext "LocalSystem" ausgeführt - somit sieht man nichts und die Programme haben die nötigen Rechte.

Nach dem Entfernen des Laufwerks das Ganze rückgängig:

[OnRemoval]

```
Letter=A open="ReMount" B: A: system=1
```

Beispiel 7:

Einem USB-Stick den Laufwerksbuchstaben wegnehmen, Dateisystem mittels <u>EjectMedia</u> abmelden ("dismount"), mittels <u>DD für Windows</u> ein Image aufspielen und zuletzt das USB-Laufwerk mittels <u>RemoveDrive</u> "sicher entfernen":

[DriveLetters] Letter=-

[OnArrival]

Letter=-

open="C:\Tools\EjectMedia.exe" %VolumeName% -D wait=1000

 $open="C:\Tools\DD.exe"-if=C:\usb-image1.bin-of=\label{limit} of=\label{limit} wait=600000$

open="C:\Tools\RemoveDrive.exe" %VolumeName% -L

Zum direkten Schreiben in ein Volume sind Admin-Rechte erforderlich, ab Vista mit aktiver UAC also noch ein elevated2=1 oder system2=1 setzen, mit letzterem sieht man aber das Konsolenfenster nicht.
Will man statt in das Volume auf die Disk schreiben, ist das Ziel für DD
\\.\%PhysicalDrive%DeviceNumber%

AutoRun für Disks ohne Partitionen

Ab USBDLM V4.8 gibt es Events für Disks, analog zu den oben beschriebenen: [OnDiskArrival], [OnDiskRemovalRequest], [OnDiskRemoval].

Beispiel:

; Festplatte die komplett als TrueCrypt-Kontainer genutzt wird auf J: mounten [On**Disk**Arrival]

DeviceType=TrueCrypt

 $open="\%ProgramFiles\%\TrueCrypt\TrueCrypt.exe"\/\q/v\Device\Harddisk\%\DeviceNumber\%\PartitionO/\Parti$

Volumespezifische Kriterien funktionieren hier naturgemäß nicht.

AutoRun beim Nutzer-Login

Die konfigurierten OnArrival-Ereignisse werden standardmäßig nur beim Anschließen eines Laufwerks

bzw. Einlegen eines Datenträgers überprüft und ggf. ausgeführt. Soll USBDLM AutoRuns auch beim Login oder Nutzerwechsel ausführen:

[Settings]
AutoRunOnLogon=1

Beim automatischen Logon ist der USBDLM-Dienst möglicherweise noch nicht gestartet, da XP nicht auf den Start aller Dienste wartet - die Illusion vom schnellen Windows-Start hat Vorrang. USBDLM verpasst so das Logon-Ereignis.

Deshalb führt USBDLM beim Dienst-Start die AutoRuns dennoch aus, wenn seit dem Systemstart weniger als zwei Minuten vergangen sind und bereits ein Nutzer eingeloggt ist, es sich also offenbar um einen verpassten Logon handelt.

Diagnose

Wenn es Unklarheiten bzgl. Kommandozeilenparameter, Arbeitsverzeichnis, Fenster-Stil usw. gibt, kann man sich mit meinem Tool TestCommandline vergewissern, dass beim gestarteten Prozess alles wie konfiguriert ankommt:

http://www.uwe-sieber.de/files/testcommandline.zip

Standardmäßig schließt es sich nach 10 Sekunden. Durch Klick auf den Countdown startet der Timer von vorn. Mit dem Kommandozeilenparameter -t:xx kann ein anderer Timeout in Sekunden festgelegt werden.

[OnArrival] open="testcommandline.exe" /open1 openstyle=max wait=2000

open="testcommandline.exe" /open2 openstyle=max

Das Fenster von TestCommandline wird <u>nicht</u> aktiv und ignoriert den Fenster-Stil, zeigt in aber an.

Wenn ein Programm versteckt oder im Kontext "System" gestartet wird, sieht man davon nichts. Die Ausgaben gestarteter Konsolenprogramme fängt USBDLM ab V4.7.2 in diesen Fällen ein und schreibt sie in die Logdatei, standardmäßig mit LogLevel=3.

Ein anderer Loglevel kann eingestellt werden, z.B. so:

[Settings]
RedirectLogLevel=2

AutoRun beim und nach dem Entfernen

Analog zur AutoRun-Funktion kann USBDLM beim "Vorbereiten zum sicheren Entfernen" sowie nach dem Entfernen eines Laufwerks eine Kommandozeile ausführen.

Um wirksam zu werden, muss das betreffende Gerät neu angeschlossen oder der USBDLM-Dienst neu gestartet werden.

1. Beim Vorbereiten zum sicheren Entfernen

Wird ein USB-Laufwerk über "Hardware sicher entfernen" abgemeldet, hat USBDLM noch Gelegenheit, auf diese Anfrage zu reagieren, während das Laufwerk noch verfügbar ist.

Zu lange sollte das Ausführen des Kommandos nicht dauern, die maximale Dauer die USBDLM zugesteht beträgt 10 Sekunden. Ist der gestartete Prozess dann noch nicht beendet, wird die Anfrage zum "Sicheren Entfernen" durch USBDLM zurückgewiesen, Windows meldet dann "USBDLM verhindert das Entfernen". XP würde bis zu 30, Vista und Win7 bis zu 15 Sekunden warten.

Beispiele

Kopieren der Datei c:\text.txt in das Verzeichnis \backup auf dem zu entfernenden Laufwerk (die 10 ist wie immer eine beliebige Zahl von 1 bis 99):

[OnRemovalRequest]

open="%windir%\System32\cmd.exe"/c copy "C:\test.txt" %drive%\backup

Das Gleiche, aber nur wenn Laufwerk X: "sicher entfernt" wird

[OnRemovalRequest]

Letter=X

open="%windir%\System32\cmd.exe" /c copy "C:\test.txt" %drive%\backup

Wie oben, aber nur wenn die Datei \backup\test.txt bereits auf dem Laufwerk vorhanden ist

[OnRemovalRequest]

FileExists=%drive%\backup\test.txt

open="%windir%\System32\cmd.exe" /c copy "C:\test.txt" %drive%\backup

Ein TrueCrypt-Laufwerk auf T: unmounten, wenn die Datei geheim.tc auf dem zu entfernenden Laufwerk vorhanden ist:

[OnRemovalRequest]

FileExists=%drive%\geheim.tc open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /d T

Leider muss man den Laufwerksbuchstaben des TrueCrypt-Laufwerks kennen. Anhand des TrueCrypt-Containers (also Partition oder Datei) unmounten wäre technisch kein Problem - leider ist das bei TrueCrypt nicht vorgesehen.

Bei langwierigen Operationen ist es besser, eine Batch-Datei zu schreiben, die alles Nötige erledigt und abschließend das Laufwerk zum sicheren Entfernen vorbereitet - dafür gibt's mein Kommandozeilentool RemoveDrive:

http://www.uwe-sieber.de/drivetools.html#removedrive

Um ein laufendes Programm zu schließen, das auf das zu entfernende Laufwerk zugreifet, kann man das ab XP verfügbare Kommando tskill nutzen.

Beispiel zum Beenden eines Prozesses, der auf das Laufwerk zugreift und etwas warten, bis das erledigt ist:

[OnRemovalRequest] Letter=X open=tskill MeinProzess afterdelay=2000

Einen Prozess abzuschießen ist aber nicht nett - der freundliche Weg ist, eine WM_CLOSE-Nachricht an das betreffende Anwendungs-Fenster zu senden.

USBDLM kann das betreffende Fenster anhand dessen Texts (Titelzeile) und (optional) dessen Fenster-Klasse finden.

Falls die zu schließende Anwendung nach Schließen des Fensters noch etwas Zeit braucht, kann USBDLM bis zu vier Sekunden warten.

Beispiel: Beim Versuch, Laufwerk X: abzumelden den Taschenrechner beenden und sicherheitshalber eine Sekunde warten:

[OnRemovalRequest] Letter=X CloseWindowText=Calc CloseWindowClass=SciCalc afterdelay=1000

Wildcards können verwendet werden, falls die Titelzeile des Fensters variabel ist:

[OnRemovalRequest] Letter=X CloseWindowText=* - Notepad CloseWindowClass=Notepad afterdelay=2000

Die Fensterklasse kann z.B. mit meinem Tool WinInfo ermittelt werden: http://www.uwe-sieber.de/files/wininfo.zip

2. Beim Fehlschlagen des Vorbereitens zum sicheren Entfernen

Wenn noch auf das zu entfernende Laufwerk zugegriffen wird, schlägt das "sichere Entfernen" fehl. Man kann dann z.B. eine gelöschte Freigabe wiederherstellen:

[OnRemovalFailed]
Letter=U
open=net share Stick_U=U:\
system=1

Mit system=1 wird es im Kontext "LocalSystem" ausgeführt, so dass es die benötigten Rechte hat, auch wenn der Nutzer kein Admin ist. Zu sehen ist dann nichts, da "LocalSystem" einen eigenen Desktop hat.

3. Nach dem Entfernen

Beispiel: Laufwerk A: wurde entfernt, ein anderes Laufwerk soll auf den frei gewordenen Buchstaben gemountet werden:

[OnRemoval]

Letter=A

MountVol ist ein Kommandozeilentool das ab Windows 2000 verfügbar ist.

Ab V5.2.5 gibt es ausserdem On<u>Safe</u>Removal und On<u>Unsafe</u>Removal:

[On**Unsafe**Removal]

open="%WinDir%\System32\mplay32.exe" /play /close "%WinDir%\Media\chord.wav" windowstyle=hidden

mplay32.exe gab es nur bis XP, ab Vista brauch man zusätzliche Software, ein VB-Script oder die Powershell um eine WAV-Datei abzuspielen ohne mit einem offenen Player zu enden.

Man kann auch die USBDLM_usr.exe nutzen, um eine Message-Box anzuzeigen:

[On**Unsafe**Removal]

open="%USBDLMPATH%\USBDLM_usr.exe" -msgbox -type:0 -icon:48 -title:Warnung -text:%BusType%-Gerät '%FriendlyName%' wurde brutal entrissen

Einschränkung

Alle beschriebenen Ereignisse beim Entfernen werden nur für Laufwerke ausgelöst, die mindestens 5 Sekunden lang angeschlossen waren.

AutoRun bei System-Ereignissen

Analog zur AutoRun-Funktion kann USBDLM bei System-Ereignissen eine Kommandozeile ausführen.

Die unterstützten Ereignisse sind:

- OnServiceStart Dienst-Start, vor dem Überprüfen der Laufwerksbuchstaben
- OnServiceStarted Dienst-Start, nach dem Überprüfen der Laufwerksbuchstaben
- OnServiceStop Dienst-Stop
- OnServiceShutdown Dienst-Shutdown (beim Herunterfahren von Windows)
- OnUserLogon Nach Nutzer-Anmeldung
- OnUserLogoff
 Nach Nutzer-Abmeldung
- OnUserLock Nach "Sperren der Arbeitsstation"
- OnUserUnlock Nach "Entsperren der Arbeitsstation"
- OnUserConnect "Verbinden" von Maus und Tastatur mit dem Desktop (z.B. nach Anmeldung)
- OnUserDisconnect "Trennen" von Maus und Tastatur vom Desktop (z.B. nach Abmeldung, Arbeitstation sperren)
- OnUserActivated Benutzer hat sich erstmals oder erneut angemeldet (Benutzer-Wechsel)
- OnUserDeactivated Benutzer hat sich abgemeldet, die Arbeitsstation gesperrt oder einen Benutzer-Wechsel eingeleitet
- OnSleepRequest Beim Standby und Ruhezustand, es wird aber nicht auf Fertigstellung gewartet; nicht ab Vista/Win7
- OnSleep Nach Auslösen von Standby und Ruhezustand, praktisch u.U. erst beim Aufwachen...
- OnResume Nach Aufwachen aus Standby und Ruhezustand
- OnUndockRequest Beim "Abdocken" über das Startmenü
- OnUndock
 Nach erfolgtem "Abdocken" über das Startmenü
- OnDock Nach erfolgtem "Andocken", aber nur wenn das "Abdocken" über das Startmenü erfolgte

OnUserConnect und OnUserDisconnect meint das "Verbinden" von Maus und Tastatur mit dem Desktop, z.B. nach An- und Abmelden eines Nutzers.

Unter XP mit aktiver "Schneller Benutzerumschaltung" wird Disconnect erst gemeldet wenn ein neuer Nutzer angemeldet wird - ist also nicht wirklich sinnvoll einzusetzen.

Ab Vista werden OnUserLock und OnUserUnlock nur dann getriggert, wenn die wirklich "Arbeitsstation sperren" gewählt wird, unter XP dagegen (fast) immer:

OnUserUnlock wird in folgender Situation unter XP nicht getriggert: Mit "Schneller Benutzerumschaltung"

wird mit "Benutzer wechseln" der aktive Nutzer gesperrt, über die Anmeldeseite ein zweiter Nutzer angemeldet, dieser wieder abgemeldet und der erste Nutzer wieder aktiviert. Obwohl zuvor ein Lock getriggert wurde, generiert XP hier kein Unlock.

Da unter XP weder auf die Lock noch auf die Connect-Ereignisse Verlass ist, gibt's ab V4.6 OnUserActivated und OnUserDeactivated.

Diese werden USBDLM-intern anhand o.g. Ereignisse generiert, wenn ein Nutzer in irgend einer Weise aktiviert bzw. deaktiviert wird, also bei "Arbeitsstation sperren", "Benutzer wechseln", "Abmelden" gibt's ein OnUserDeactivated, und beim "Anmelden" und "erneut Anmelden" ein OnUserActivated.

Für Ereignisse bei denen der Nutzer bereits abgemeldet ist, muss ein system=1 angegeben werden, so dass das jeweilige Programm im Kontext "LocalSystem" ausgeführt wird - den Nutzerkontext gibt's ja schon nicht mehr.

Bei OnSleepRequest wartet Windows leider nicht bis die Nachricht verarbeitet wurde, sondern nur maximal zwei Sekunden. Es gibt also keine Garantie, dass irgendwelche Aktionen noch vor dem Standby oder Ruhezustand ausgeführt und vor allem fertiggestellt werden. Ab Vista gibt es OnSleepRequest nicht mehr. OnSleep wird u.U. erst beim Aufwachen ausgeführt.

Wenn die Windows-Option "Kennwort beim Reaktivieren aus dem Standbymodus anfordern" aktiviert ist, wird unter XP die Arbeitsstation <u>beim Aufwachen</u> gesperrt, dementsprechend kommen OnUserLock, OnUserDisconnect und OnUserDeactivated erst dann. USBDLM kann das Sperren schon beim Standby veranlassen, dann kommen auch die Ereignisse vor dem Einschlafen:

[Settings]
LockWorkstationOnSleep=1

Hundertprozentig zuverlässig ist das aber nicht.

Beispiel fürs Anlegen eines Netzlaufwerks nach Nutzer-Login:

[OnUserLogon] open=net user X: \\server\\freigabe /user:nutzername passwort openstyle=min

Die sonstigen, von [OnArrival] bekannten Parameter, wie openstyle, system usw. sind auch hier gültig, mehrere open-Zeilen können durch Anhängen einer Ziffer 1..9 genutzt werden. Ab V5 geht's auch ohne Ziffern, eine open-Zeile beginnt dann einen neuen Parametersatz.

Ab V4.5 sind mehrere Abschnitte eines Typs möglich, die einzigen Kriterien sind die des <u>aktiven Nutzers</u>, ein laufender <u>Prozess</u> und die für das <u>Vorhandensein einer Datei</u>.

Beispiel zum Start des WIA-Dienstes für Admins und zum Stoppen desselben für Nicht-Admins beim Login:

[OnUserLogon]
IsUserAdmin=1

open=net start stisvc system=1

[OnUserLogon] IsUserAdmin=0 open=net stop stisvc system=1

system=1 bewirkt, dass die NET.EXE im Kontext "LocalSystem" ausgeführt wird. Somit sieht der Nutzer das eigentlich aufpoppende Konsolenfenster nicht und der Nicht-Admin hat kein Problem mit fehlenden Rechten.

AutoRun für andere Geräte

Analog zur AutoRun-Funktion für Laufwerke kann USBDLM beim Anschließen und Entfernen beliebiger Geräte eine Kommandozeile ausführen.

Einziges gerätespezifisches Kriterium ist die Geräte-ID, es funktioniert ausschließlich die "Geräteinstanzkennung". Diese findet man im Geräte-Manager:

Das betreffende Gerät mit rechts anklicken -> Eigenschaften, Reiter "Details". Hier ist unter XP schon die "Geräteinstanzkennung" ausgewählt. Dann unten auf die Geräte-ID klicken und "Strg+C" drücken - das kopiert sie in die Windows Zwischenablage. Ein Kontextmenü mit "Kopieren" gibt es erst ab Vista.

XP vor SP2 zeigt den Reiter "Details" nur wenn die Umgebungsvariable DEVMGR_SHOW_DETAILS auf 1 gesetzt ist.

Diese REG-Datei setzt den Wert: <u>devmgr show 1.reg</u> Wirksam wird das nach dem nächsten Login.

Ab V4.7.1 sind weitere verfügbare Kriterien jene, die nicht im Zusammenhang mit einem neuen Laufwerk stehen, also <u>Dateien</u>, <u>Nutzer</u>, <u>Computername</u>, <u>Prozess</u>, <u>Betriebssystem</u> und <u>Zeit</u>.

Beispiel für das Starten des Taschenrechners beim Anschließen eines bestimmten USB-Gerätes:

[OnDeviceArrival]
DeviceID1=USB\VID_090C&PID_1000\A74000000000097
open=calc

Das Selbe nach dem Entfernen:

[OnDeviceRemoval]
DeviceID1=USB\VID_090C&PID_1000\A74000000000097
open=calc

[OnDeviceRemovalRequest] gibt es nicht.

Die sonstigen, von [OnArrival] bekannten Parameter, wie openstyle, system usw. sind auch hier gültig.

Standardmäßig registriert sich USBDLM nur für Gerätebenachrichtigungen der Geräteklassen USB, DISK, CDROM, FLOPPY und VOLUME. Wenn ein Abschnitt OnDeviceArrival oder OnDeviceRemoval vorhanden ist, muss es sich für alle Geräteklassen registrieren. Bei Standby/Ruhezustand und beim Aufwachen daraus rauscht dann eine Flut von Benachrichtigungen herein und für jede wird nach einem

passenden Abschnitt gesucht...

Balloontips

Bis einschließlich Windows 7 hat Windows nur für erstmals angeschlossene Laufwerke einen Ballontip angezeigt, aber ohne Angabe des zugeordneten Laufwerksbuchstabens. Hier hilft USBDLM aus. Ab Windows 8 zeigt auch Windows einen Balloontip eines neu angeschlossenen Laufwerks, aber nur wenn auch ein Laufwerksbuchstaben zuordnet wurde. Bei NTFS-Mountpunkten kommt da nichts.

Ab V5.0 können die USBDLM-Balloontips an die Wünsche des Nutzers angepasst werden und so z.B. auch die Datenträgerbezeichnung, den freien Speicherplatz und vieles andere Anzeigen.

Nach sechs Sekunden oder beim Klick auf den Balloontip wird dieser geschlossen, es kann auch eine Aktion bei Klick festgelegt werden, siehe weiter unten.

[BalloonTips]
Enabled=1
EnabledOnRemoval=1
Timeout=6000

Wird die Maus über dem Balloon-Tip bewegt, beginnt die Timeout-Zeit von vorn.

Der Text wird aus wenigen Text-Bausteinen zusammengesetzt. Texte sind in deutsch und einigen anderen Sprachen enthalten und werden abhängig von der in Windows eingestellten Sprache benutzt.

Man kann sich die Texte auch anpassen, folgende Text-Bausteine sind enthalten:

[BalloonTips]
Text_Drives=Laufwerke
Text_Partitions=Partitionen
Text_Media=Medium
Text_mounted_to=bereitgestellt unter
Text_and=und
Text_removed=Entfernt
Text_safely_removed=Sicher entfernt
Text_unsafely_removed=Unsicher entfernt
Text_ejected=Ausgeworfen
Text_NetworkDrive=Netzlaufwerk
Text_SubstDrive=Subst-Laufwerk

Ergibt dann z.B. für

- ein Laufwerk mit einem Volume, z.B. ein USB-Stick oder eine Festplatte mit einer Partition bereitgestellt unter X:\
- eine Festplatte mit mehreren Partitionen

- 2 Partitionen bereitgestellt unter X:\ und Y:\
- ein Gerät mit mehreren Laufwerken, z.B. ein Multislot-Cardreader
 3 Laufwerke bereitgestellt unter X:\, Y:\ und Z:\
- ein eingelegtes Medium, z.B. eine Flashkarte falls der Mountpunkt gerade zugeordnet wurde Medium bereitgestellt unter X:\

Die maximale Länge pro Text-Baustein ist 47 Zeichen.

Wenn sich in einer Fremdsprache für "bereitgestellt unter" die Singular- und Plural-Varianten unterscheiden, kann für Plural der Eintrag

Text mounted to plr konfiguriert werden.

Viele verschiedene Varianten sind hier beim Zusammenbasteln der Balloontips berücksichtigt. Daher sind frei konfigurierbare Balloon-Texte nicht mit vertretbarem Aufwand machbar. Für einfache Laufwerke mit einem Volume und einem Buchstaben ist es aber kein Problem:

Beispiel für Laufwerksbuchstabe, Volume-Label ("Datenträgerbezeichnung"), Dateisystem und USB-Geschwindigkeit:

Text_SingleDrive=bereitgestellt unter %Root% "%VolumeLabel%" [%FsName%] - %UsbSpeed%

ergibt z.B.

bereitgestellt unter X:\ "Backup" [NTFS] - SuperSpeed

%Root%, "%VolumeLabel%", %FsName% und %UsbSpeed% sind <u>USBDLM-Variablen</u>.

Ab V5.0 hat man noch mehr Möglichkeiten, die Balloontips anzupassen, siehe Balloon-Tip nutzerdefiniert.

Um "%VolumeLabel%" [%FsName%] im oberen Beispiel rechtsbündig zu bekommen, kann ein "Zentrum" definiert werden. Der Platzhalter dafür ist "!! ". Der wird zunächst durch sechs Leerzeichen ersetzt. Ergibt sich aus der Titelzeile des Balloons ein breiterer Balloon und somit mehr Platz, werden es entsprechend mehr. Soll etwas Abstand von rechts eingehalten werden, funktionieren angehängte Leerzeichen nicht - die werden beim Einlesen der INI-Datei ignoriert. Man nehme stattdessen Unterstriche, die werden am Anfang und am Ende durch Leerzeichen ersetzt.

Beispiel:

Text_SingleDrive=bereitgestellt unter %Root% !! "%VolumeLabel%" [%FsName%] - %UsbSpeed%____

Für neu eingelegte CDs/DVDs gibt es Text_singleCdDvd und für Wechseldatenträger Text_singleMedia. Ein Balloontip wird aber nur dann angezeigt, wenn beim Einlegen ein Mountpunkt zugeordnet wird, also NoMediaNoLetter für das Laufwerk aktiv ist.

Wenn NoMediaNoLetter=1 konfiguriert ist, kann ein Balloontip schonmal so aussehen:

bereitgestellt unter ---, ---, --- und ---

Auf den ersten Blick überflüssig, informiert es darüber, dass der Reader vier Laufwerke beherbergt und allen vieren kein Laufwerksbuchstabe zugeordnet wurde.

Ab V5.0 gibt's optional eine Kurznotation falls (dank <u>NoMediaNoLetter</u>) maximal einem Slot ein Mountpunkt zugeordnet wurde:

[BalloonTips]
MscrCompact=1

Mehrere Mountpunkte

Ab V4.3 kann USBDLM mehrere Mountpunkte zuordnen. Standardmäßig werden in Balloontips maximal zwei pro Volume angezeigt. Wer mehr braucht:

[BalloonTips]
MaxMountPointsToShow=3

Oder wer nur den ersten sehen will:

[BalloonTips]
MaxMountPointsToShow=1

Standardmäßig wird der Laufwerksbuchstabe (falls vorhanden) als erster gezeigt. Wer es umgekehrt haben will:

[BalloonTips]
MountPointsReverse=1

Zeichensatz

Ab V4.6 arbeitet USBDLM mit UTF-16 - es gibt es bzgl. Zeichensatz nichts mehr einzustellen. Bei Problemen auch die USBDLM.INI in UTF-16 speichern, Notepad nennt es "Unicode".

"Rechts-nach-links"-Schriften werden derzeit nicht unterstützt.

Einstellungen für Balloontips

Windows-Balloontips bei Bedarf unterdrücken

Wenn ein Laufwerk zum ersten mal angeschlossen wird, zeigt Windows wenig hilfreiche Balloontips. USBDLM räumt die bei Bedarf aus dem Weg oder deaktviert sie vorübergehend:

[BalloonTips]
SuppressWindowsBalloons=1

- 0: Aus
- 1: Sofort schließen bei Erscheinen, sichtbar für einen kurzen Moment (Default V4.x und ab V5.1.4)
- 2: Vorübergehend deaktivieren (Default V5.0 bis V5.1.3)

Das Windows "Hardware sicher entfernen" Tray-Icon reagiert nicht gut auf das Deaktivieren von Balloontips - es funktioniert nicht mehr nachdem es eigentlich einen Balloontip angezeigt hätte.

Windows 10 Stil

Ab Windows 10 sehen Balloontips anders aus (und heißen "toast notifications), USBDLM versucht das nachzuempfinden. Unter Windows 8 hatten nur die Balloontips für neue Laufwerke das neue Design, andere Balloontips im Info-Bereich sahen aus wie unter Windows 7.

Standardmäßig zeigt USBDLM unter Windows 10 die Balloontips im neuen Stil. Falls man es klassisch haben möchte:

[BalloonTips] Win10Style=0

Die Farben werden dabei aus HKCU\Software\Microsoft\Windows\DWM\AccentColor abgeleitet, wer es besser weiß, kann die Farben selbst festlegen:

[BalloonTips]
BackColor=0xA02020
CaptionColor=0xFFFFFF
TextColor=0xFFCFCF

Auf das letzte Laufwerk warten

Wenn ein Gerät mit mehreren Laufwerken angeschlossen wird, z.B. eine Festplatte mit mehreren Partitionen oder ein Multislot-Cardreader, kann USBDLM mit dem Ballon-Tip warten, bis alle Laufwerke bereit sind. Normalerweise wird der Balloon-Tip für das erste Laufwerk angezeigt und dann für jedes weitere aktualisiert.

[BalloonTips]
WaitForLastVolume=1

Laufwerksgröße anzeigen

[BalloonTips] ShowSize=1

So wird hinter dem Geräte-Namen noch die Größe des Laufwerks angezeigt, z.B.

Corsair Flash Voyager USB Device - 8 GB

Laufwerksgröße runden

Bei Größen zwischen 8 und 1000 MB kann USBDLM die Größe so runden, dass immer gängige Größen wie 256 und 512 MB statt krumme Werte wie 262 und 509 MB angezeigt werden. Außerdem werden immer ganze Gigabytes angezeigt. Ab V5.0 ist das standardmäßig aktiv, falls man es nicht mag:

[BalloonTips]
RoundSize=0

Icon mit USB-Geschwindigkeit

Ab V5.0 wird bei USB-Laufwerken ein Icon mit FS, HS, SS oder SS+ für Full-, High-, Super und SuperSpeedPlus gezeigt. Hierbei handelt es sich um die tatsächliche Verbindungsgeschwindigkeit. SuperSpeedPlus kann erst ab Windows 10 angezeigt werden. Wenn man das nicht will:

[BalloonTips]
UsbSpeedIcon=0

Anzeige der vom USB-Gerät maximal unterstützten USB-Geschwindigkeit oder der USB-Version im Balloon-Titel

Ab V5.0 wird statt nur "USB Device" die USB-Version mit angezeigt, z.B. "USB 3.0 Device". Alternativ kann auch die maximale vom Gerät unterstützte USB-Geschwindigkeit angezeigt werden, z.B. "USB SuperSpeedPlus Device":

[BalloonTips]
UsbSpeedTitle=2

0 -> aus

1 -> maximale Geschwindigkeit als Text

2 -> Version als Nummer (Default)

Keine Balloontips für TrueCrypt-Volumes

Ab V4.3 zeigt USBDLM auch BalloonTips für gemountete TrueCrypt-Volumes. Soll es das nicht tun:

[BalloonTips]
ShowTrueCrypt=0

Balloontips für Netz- und Subst-Laufwerke

Ab V4.6 kann USBDLM auch BalloonTips für neu angelegte "virtuelle" Laufwerke anzeigen. "Virtuell" mein hier, dass zwar ein Laufwerksbuchstabe vorhanden ist, sich dahinter aber kein lokales "Speichervolumen" verbirgt.

Das sind insbesondere Netz- und Subst-Laufwerke. Ab V5.0 standardmäßig aktiv. Falls nicht gewollt:

[BalloonTips]
ShowVirtual=0

Ab Vista / Windows 7 kommt noch die User Account Control (UAC) ins Spiel - hier ist der Nutzer-Kontext aufgeteilt in einen eingeschränkten und einen nicht eingeschränkten ("Als Administrator ausführen"). Um Balloontips auch für Netz- und Subst-Laufwerke zu zeigen, die im nicht eingeschränkten Kontext angelegt werden, konfiguriert man 2 statt 1. USBDLM zeigt dann im Balloon-Tip-Titel zusätzlich "(elevated)" wenn das Laufwerk für den nicht eingeschränkten Nutzer kam und "(normal+elev)" falls für beide (z.B. bei TrueCrypt-Volumes oder bei Netzlaufwerken wenn EnableLinkedConnections aktiviert ist, " (normal+elev)".

Die Texte kann man anpassen (auch auf leer setzen):

[BalloonTips]
Text_ContextNormal=normal
Text_ContextElevated=elevated
Text_ContextBoth=normal+elevated

Balloontips für U3-Sticks

Standardmäßig zeigt USBDLM im Balloontip nur das Daten-Laufwerk an, da das U3-CDROM ja nur das Autorun-Vehikel ist. Will man dennoch beide Laufwerke im Balloontip angezeigt haben:

[BalloonTips] ShowU3CDROM=1

Das Daten-Laufwerk wird als erstes angezeigt.

Keine Balloontips für bestimmte Laufwerke

Immer mehr in Mode kommen USB-Geräte wie Drucker, WebCams oder WLAN-Sticks mit einem "Selfinstaller". Das ist ein gefälschtes CD-ROM-Laufwerk über das automatisch Treiber installiert werden sollen. Meist deaktivieren sich diese CD-ROMs ein paar Minuten nach dem Anschließen. Das ist bestens

geeignet, um auf Rechnern ohne USBDLM die Laufwerksbuchstaben durcheinanderzubringen. Damit USBDLM dafür keinen Balloontip zeigt, muss man das anhand dessen Hardware-Geräte-ID oder des "Friendly Name" sperren. Beispiel:

[BalloonTipsExclude]
DeviceID=USB\VID_057C&PID_62FF
DeviceID=selfinstall

Die ID muss mindestens acht Zeichen lang sein oder <u>Wildcards</u> nutzen. Man holt sie sich mit dem mitgelieferten UsbDriveInfo, siehe <u>Buchstaben abh.</u> von der <u>Geräte-ID</u>.

Aktionen beim Klick auf den Balloontip

Analog zu AutoRun können Ereignisse bei Links-, Rechts- und Mittel-Klick definiert werden.

;Explorer bei Links-Klick [OnBalloonClick] open="%windir%\explorer.exe" %drive%

;Foto-Software bei Rechts-Klick
[OnBalloon**R**Click]
open="C:\Programme\FotoSoft\fotosoft.exe" %drive%

Es können mehrere Ereignisse abhängig von Laufwerks-Eigenschaften definiert werden, genau wie für [AutoRun] beschrieben.

Wenn im Balloontip mehr als ein Laufwerk angezeigt wird, gilt das Ereignis nur für das erste Laufwerk mit einem Laufwerksbuchstaben.

Unter XP öffnet explorer /e ein Explorer-Fenster mit Laufwerks-Baum, das sollte hier nicht benutzt werden, da der Explorer ca. 4 Sekunden braucht, um ein neues Laufwerk in den Laufwerks-Baum einzutragen. Wenn man vorher auf ein neues Laufwerk zugreift, bekommt man u.U. einen leeren Eintrag im Laufwerksbaum.

Nutzerdefinierte Balloontips

Einstieg

Ab USBDLM V5.0 kann man analog zu Abschnitten wie DriveLetters oder OnArrival solche zum freien Festlegen der Texte in Balloontips anlegen, allerdings nur für Laufwerke mit genau einem Volume.

Folgende Abschnitte werden unterstützt:

[BalloonTipsOnArrival]
 Volume angeschlossen

• [BalloonTipsOnRemoval] Volume entfernt (sicher/unsicher unbekannt)

• [BalloonTipsOnSafeRemoval] Volume sicher entfernt

• [BalloonTipsOnUnsafeRemoval] Volume unsicher entfernt

• [BalloonTipsOnMediaArrival] Datenträger eingelegt (Mountpunkt war schon vorhanden)

• [BalloonTipsOnSafeMediaRemoval] Datenträger entfernt/ausgeworfen und war bereits dismountet

• [BalloonTipsOnUnsafeMediaRemoval] Datenträger entfernt/ausgeworfen und war noch gemountet

Title legt den fett gedruckten Titel des Balloontips fest, Text den Text, in beiden können <u>USBDLM-Variablen</u> verwendet werden.

Mit Icon kann eines der <u>USBDLM-internen Icons</u> festgelegt werden, eine ICO-Datei oder eine Icon-Ressource einer EXE, DLL etc. Auch wenn die Notation so aussieht wie oft für den Windows Explorer in der Registry anzutreffen, funktioniert der Icon-Index hier anders (ohne Gruppen), er entspricht dem was z.B. der <u>ResHacker</u> anzeigt.

Beispiele

Beim Anschließen

; TrueCrypt-Volumes mit Größe, Name des Dateisystems und dem blauen Schlüssel-Icon aus der

TrueCrypt.exe

[BalloonTipsOnArrival]

DeviceType=TrueCryptVolume

Title=TrueCrypt Volume - %VolumeSize%

Text=bereitgestellt unter %Root% (%FsName%)

Icon="%ProgramFiles%\TrueCrypt\TrueCrypt.exe",2

; USB-Laufwerke mit Name, USB-Version, maximaler Geschwindigkeit (z.B. SuperSpeed), tatsächlicher Geschwindigkeit und Laufwerksgröße

; (das Kriterium BusType=USB ist auch hier Default, deshalb gilt der Abschnitt nur für USB-Laufwerke) [BalloonTipsOn**Arrival**]

Title=%FriendlyName% %BusType% %UsbVersion% %UsbMaxSpeed% Device at %UsbSpeed% - %DriveSize%

Text=gemountet auf %Root% (%FsName%)

Beim Entfernen

; TrueCrypt-Volumes mit Größe, Name des Dateisystems und dem orangen Schlüssel-Icon aus der TrueCrypt.exe

[BalloonTipsOn**Removal**]

DeviceType=TrueCryptVolume

Title=%FriendlyName% - %VolumeSize%

Text=entfernt: %Root% (%FsName%)

Icon="C:\Program Files\TrueCrypt\TrueCrypt.exe",6

Beim sicheren Entfernen

; alle BusTypes

[BalloonTipsOnSafeRemoval]

BusType=ANY

Title=%FriendlyName% %BusType% Device - %UsbSpeed% - %DriveSize%

Text=sicher entfernt von %Root% !! %FsName%

Beim unsicheren Entfernen

; alle BusTypes

[BalloonTipsOnUnsafeRemoval]

BusType=ANY

Title=%FriendlyName% %BusType% Device - %UsbSpeed% - %DriveSize%

Text=überraschend entfernt von %Root% !! %FsName%

Beim Einlegen eines Datenträgers

;USB, Kamera-Icon aus der Windows7 shell32.dll

[BalloonTipsOn**MediaArrival**]

Title=%FriendlyName% %BusType% Device - %DriveSize%

Text=Flashkarte eingelegt: %Root% (%FsName%)

Icon=C:\Windows\System32\shell32.dll,1347

Beim sicheren Entnehmen/Auswerfen eines Datenträgers

;USB

[BalloonTipsOn**SafeMediaRemoval**]

Title=%FriendlyName% %BusType% Device - %DriveSize% Text=Flashkarte sicher entfernt: %Root% (%FsName%)

Beim unsicheren Entnehmen/Auswerfen eines Datenträgers

;USB

[BalloonTipsOn**UnsafeMediaRemoval**]
Title=%FriendlyName% %BusType% Device - %DriveSize%
Text=Flashkarte überraschend entfernt: %Root% (%FsName%)

Variablen

Bei der Konfiguration von AutoRun, Explorer Label und Mounten in NTFS-Ordner können einige Variablen benutzt werden:

Variablen

```
Variable
             Beschreibung
                                Beispiel
%DriveLetter%
                Laufwerksbuchstabe
                                     X
%Drive%
              Laufwerk
                               X:
%Root%
              Stammverzeichnis
                                  X:\
                                   Corsair Flash Voyager
%DevName%
                 Gerätename
%FriendlyName%
                                    Corsair Flash Voyager
                  Gerätename
%BusType%
                Anschluss-Typ
                                    USB
%DriveDeviceId%" Geräte-ID des Laufwerks USBSTOR\DISK&VEN_CORSAIR&PROD_FLASH...
%CtrlDeviceId%"
                 Geräte-ID des Kontrollers USB\VID_090C&PID_1000\...
                                    123456789 oder ---
%UsbSerial%"
                USB serial number
%UsbPortName%
                  USB Port-Name
                                       3-1-1
                                       High-Speed
%UsbSpeed%
                USB-Geschwindigkeit
                  max. USB-Geschw. des LW Super Speed
%UsbMaxSpeed%
%UsbVersion%
                USB-Version
                                   2.0
              Datenträgerbezeichnung Mein Stick
%Label%
                 Datenträgerbezeichnung Mein Stick
%VolumeLabel%
%LetterFromLabel% Buchstabe aus Bez.
                                       X wenn Label z.B. Laufwerk X ist
%DriveType%
                Laufwerkstyp
                                   REMOVABLE
%FsName%
                Datei-System
                                  FAT32
%Size%
             Größe des Volumes
                                  2 GB
%VolumeSize%
                 Größe des Volumes
                                      2 GB
%DiskSize%
               Größe des Datenträgers
                                     20 GB
               Größe des Datenträgers 20 GB
%DriveSize%
%KernelName%
                 Kernel-Name
                                     \Device\Harddisk3\DP(1)0-0+d
%PartitionName%
                 Partitions-Name
                                     \Device\Harddisk2\Partition1
%DeviceNumber%
                  Geräte-Nummer
                                       2
%PartitionNumber% Partitions-Nummer
%VolumeName%
                  Volume-Name
                                      \\?\Volume{GUID}\
%PureVolumeName% purer Volume-Name
                                          Volume{GUID}
%VolumeNameGuid% GUID des Volume-Namens {GUID}
              subst/net/TC target
                                 \\MyServer\public
%Target%
```

Das waren die die sich auf ein Volume beziehen, also bei OnArrival, OnRemoval usw. mit den Daten des gerade behandelten Volumes gefüllt sind.

Weitere Variablen sind:

%Date% Datum (yymmdd) 130331 %Date4% Datum (yyyymmdd) 20130331 %DateISO% Datum (yyyy-mm-dd) 2013-03-31

%Time% Zeit (hh:mm:ss) 12:00:00

%UserName% Nutzer-Name Administrator

%UsbdlmPath% USBDLM-Ordner C:\Tools\USBDLM

Außerdem können alle Variablen gestarteten Prozessen als Umgebungsvariable zur Verfügung gestellt werden

Bis einschließlich V4.2 war das eine globale Einstellung:

[Settings]

UsbdlmVariablesToOpenEnvironment=1

Ab V4.3 ermittelt USBDLM nur noch Laufwerks-Infos die wirklich benötigt werden. Deshalb muss man jetzt konfigurieren, welche Variablen benötigt werden.

"=1" geht auch, aber damit sind alle Optimierungen wirkungslos.

Beispiel:

[Settings]

UsbdlmVariablesToOpenEnvironment=DevName, FsName, VolumeLabel

UsbDriveInfo kann die Variablen anzeigen (Options -> Show USBDLM Variables).

Wildcards

Bei der Konfiguration der Kriterien DeviceID, VolumeLabel, UsbPortName, UserName und UserGroup können Wildcards (=Platzhalterzeichen) benutzt werden.

Dabei steht der Stern '*' für Null bis n Zeichen, das Fragezeichen '?' für genau ein Zeichen.

Buchstaben werden unabhängig von Groß/Kleinschreibung ausgewertet.

	passt z.B. für		
Backup?	Backup1, Backup2, BackupA, BackupB +	Backup, B 	ackup01
Backup*	Backup, Backup1, Backup01, Backup001	Back 	
*Backup	Backup, MyBackup	Backup1	
ck	Backup, Bäcker, ck +	Akku	
B*up	Backup, Buckup, Bup +) Bäcker	
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_090 +	C&PID_1111.	USB\VID_2222
admin*	Admin, Administrate 	or adam	

USB Schreibschutz

Seit Windows XP mit SP2 kann Windows für USB-Laufwerke einen Schreibschutz aktivieren. Leider ist das eine globale, nicht gerätespezifische Einstellung.

Ab V4.6.9.2 kann USBDLM diese Einstellung abhängig vom Gerät vornehmen, so dass sie im Moment des Mountens des Volumes wunschgemäß gesetzt ist.

Hintergrund-Infos:

Beim Anschließen eines USB-Laufwerks meldet Windows zunächst das USB-Gerät, dann die Disk und zuletzt das "Speichervolumen", das was typischerweise einen Laufwerksbuchstaben bekommt. Zwar wirkt der Schreibschutz nur auf das Volume, um aber sicherzustellen, dass der Registry-Eintrag rechtzeitig gesetzt wird, erledigt USBDLM das bereits beim Ankunft des USB-Gerätes. Somit ist das einzig verfügbare Kriterium die Geräte-ID des USB-Gerätes.

Die Einstellung befindet sich im Registry Wert "WriteProtect" (Typ DWORD) unter HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies. 1 aktiviert den Schreibschutz, 0 deaktiviert ihn.

USBDLM kann abhängig vom Gerät eine 1 oder 0 schreiben. Falls ein Abschnitt UsbWriteProtect oder UsbNoWriteProtect konfiguriert ist, benötigt USBDLM für alle anderen USB-Laufwerke einen Standardwert im Abschnitt Settings.

Beispiel:

; Standard: setze eine 0 für alle Geräte die weder im Abschnitt ; [UsbWriteProtect] noch in [UsbNoWriteProtect] konfiguriert sind [Settings] UsbWriteProtect=0

; setze eine 1 für diese Geräte [UsbWriteProtect] DeviceId1=USB\VID 13FE&PID 1F00

; setze eine 0 für diese Geräte [Usb**No**WriteProtect] DeviceId1=USB\VID_0951&PID_1624

Die IDs können auch in einer externen Textdatei hinterlegt werden, so wie für das Kriterium Geräte-ID gezeigt.

Typische Szenarien:

• alle USB-Laufwerke per UsbWriteProtect=1 mit Schreibschutz, Ausnahmen per Abschnitt [Usb**No**WriteProtect]

• alle USB-Laufwerke per UsbWriteProtect=0 ohne Schreibschutz, Ausnahmen per Abschnitt [UsbWriteProtect]

Der Standardwert wird auch gesetzt wenn der USBDLM-Dienst gestoppt wird.

Laufwerke die beim Systemstart bereits angeschlossen sind, sind bereits eingebunden und können somit nicht mehr behandelt werden. Für diese Laufwerke war dann der Standardwert wirksam.

Der ganze Mechanismus ist nicht 100% zuverlässig wenn mehrere USB-Laufwerke gleichzeitig angeschlossen werden und für diese unterschiedliche Einstellung für den Schreibschutz gemacht werden sollen.

Passwortabfrage

Ab V4.5 kann USBDLM Laufwerksbuchstaben zunächst entfernen und nach einen Passwort fragen.

Beispiel:

; eingeschränkte Nutzer werden nach einem Passwort gefragt [Password] UserGroup=Benutzer password=geheim

; Administratoren fallen oben durch, brauchen somit kein Passwort

Wenn das Passwort in Textform abgelegt ist, sollte der Lesezugriff für eingeschränkte Nutzer von der USBDLM.INI entfernt werden - sonst kann der Nutzer einfach nachsehen... Oder man notiert nur den MD5 Hashwert, das erschwert es zumindest, ein dazu passendes Passwort zu

[Password]

generieren:

UserGroup=Benutzer passwordMD5=E8636EA013E682FAF61F56CE1CB1AB5C

MD5-Text.zip

Der USBDLM-Dienst akzeptiert ein Passwort pro Sekunde, die Anzahl der Versuche ist nicht begrenzt.

Wenn die Passwortabfrage abgebrochen wird, wird das Laufwerk "sicher entfernt" oder dessen Datenträger ausgeworfen.

Wird das Laufwerk oder der Datenträger wieder entfernt während die Passwortabfrage offen ist, wird diese geschlossen.

Gültige Kriterien für Passwort-Abschnitte sind:

- UserName
- UserGroup
- UserIsAdmin
- DeviceID
- BusType

Im Gegensatz zu allen anderen Abschnittstypen sind hier alle Laufwerkstypen betroffen, also auch Floppys und CD/DVD-Laufwerke. Standard-Bustype ist wie immer USB.

Mit Hilfe des "USBDLM Remote Password Tool", kann das Passwort auch über's Netzwerk eingegeben werden. So braucht der Administrator dem Nutzer das Passwort ggf. nicht verraten...

<u>USBDLM-Remote.zip</u>

Der Passwort-Dialog auf dem Client-Computer muss offen bleiben, er verschwindet sobald das korrekte Passwort über's Netzwerk empfangen wurde.

Das USBDLM Remote Password Tool verlässt sich darauf, dass auf den Client-Computer zugegriffen werden kann, dass also bereits eine Anmeldung erfolgt ist. Alternativ kann USBDLM anonymen Zugriff auf seine Password-Schnittstelle erlauben:

[Settings] RemotePasswordAnonymous=1

Laufwerke am Einschlafen hindern - Keep Alive

Manche USB-Platte legt sich nach kurzer Zeit schlafen und braucht dann beim nächsten Zugriff einen Moment um wieder in Gang zu kommen.

USBDLM kann durch zyklische Lesezugriffe das Einschlafen verhindern.

Konfiguriert wird das mittels Abschnitten [DiskKeepAliveXX].

Wie bei jedem anderen nummerierbaren Abschnitt gelten die Standard-Kriterien BusType=USB und DriveTypes=REMOVABLE, FIXED - der Abschnitt ist also standardmäßig für USB-Wechseldatenträger und USB-Festplatten zutreffend, man wird also typischerweise wenigstens DriveType=FIXED oder eine/mehrere Geräte-IDs konfigurieren.

Beispiel:

; von allen USB-Festplatten alle 60 Sekunden lesen [DiskKeepAlive] DriveType=FIXED Interval=60000

; von einer bestimmten USB-Festplatte alle 60 Sekunden lesen [DiskKeepAlive]
DeviceID=USB\VID_0781&PID_5580\AA010319131128111123
Interval=60000

Standard für Interval ist Null, was das Lesen deaktiviert. Sonst wäre ein leerer Abschnitt [DiskKeepAlive] für alle USB-Laufwerke wirksam - das wäre irgendwie merkwürdig gewesen...

Auto-Dismount

Ab V5.0 kann USBDLM zyklisch versuchen Volumes zu dismounten, also das Dateisystem abzumelden. Das klappt nur, wenn keine Dateien auf diesem Volume geöffnet sind. Bei nächsten Zugriff mountet Windows das Laufwerk automatisch, was aber einen Moment dauert, insbesondere weil beim Dismounten der Lese-Cache verworfen wird.

So kann man USB-Sticks ohne "Hardware sicher entfernen" einfach abziehen. Ab V5.1 kann USBDLM ein Tray-Icon mit bis zu vier dismounteten Laufwerksbuchstaben zeigen. Auch UsbDriveInfo zeigt den Mount-Status.

Warnung: Bei einer SSD genügt das Dismounten des Dateisystems nicht! Man weiß hier nie, welche internen Optimierungsprozesse gerade ablaufen. Hier ist "Hardware sicher entfernen" Pflicht, das sendet (hoffentlich) ein Shutdown-Signal bis zur SSD.

USBDLMs Auto-Dismount funktioniert nur mit lokalen Laufwerken und z.B. nicht mit TrueCrypt-Volumes, da diese die nötigen Banachrichtigungen nicht unterstützen.

Problem:

Das Explorer-Tray-Icon "Hardware sicher entfernen" friert offenbar ein, wenn das letzte Laufwerk in dessen Menü z.B. über das Laufwerks-Kontextmenü im Windows Explorer ausgeworfen wird und dieses zuvor schon dismountet war - egal ob durch USBDLM oder wen auch auch immer.

Auto-Dismount wird ananlog zu Laufwerksbuchstaben konfiguriert. Im einfachsten Fall z.B. einheitlich für alle Wechseldatenträger (DriveType REMOVABLE):

[AutoDismount]
DriveType=REMOVABLE
Interval=10s

Mit FIXED Laufwerken klappt das ab Vista nicht so gut, hier greift der Windows SearchIndexer ohne ersichtlichen Grund regelmäßig auf die Laufwerke zu, so dass sie gleich wieder gemountet werden. Ein Möglichkeit, das zu unterbinden, habe ich keine gefunden, außer den Dienst "Windows Search" (WSearch) zu deaktivieren.

Ein Intervall von Null deaktiviert Auto-Dismount. So kann man Laufwerke davon ausschließen:

; Kein Auto-Dismount für dieses Laufwerk [AutoDismount]
DeviceID=USB\VID_13FE&PID_1F00
Interval=0

; für alle anderen USB REMOVABLE Laufwerke [AutoDismount] DriveType=REMOVABLE

Interval=10s

Nach Ankunft eines Laufwerks wird zunächst kein Auto-Dismount betrieben, der Nutzer will ja schließlich auf das Laufwerk zugreifen. Diese Gnadenfrist beträgt 30 Sekunden und kann global eingestellt werden:

[Settings]

AutoDismountGracePeriod=30s

Ab V5.1 kann USBDLM ein Tray-Icon mit bis zu vier dismounteten Laufwerken anzeigen:

[Settings]

AutoDismountIcon=2

- 0: Aus
- 1: Anzeigen wenn mindestens ein Laufwerk dismountet ist
- 2: Immer anzeigen

Standard ist 2 wenn es minidestens einen Abschnitt [AutoDismount] gibt, sonst 0.

Die Hintergrundfarbe kann eingestelt werden (Hexwert BBGGRR).:

[Settings]

AutoDismountIconBackgroundColor=0x80FF80

Bei einer Mausbewegung auf dem Icon wird sofort versucht, alle für Auto-Dismount konfigurierten Laufwerke zu dismounten - deshalb ist es sinnvoll das Icon auch ohne dismountete Laufwerke anzuzeigen. Wenn man auf die Automatik ganz verzichten will, kann man einen Interval von -1 konfigurieren, das ergibt einen Intervall von 232 Millisekunden (49,7 Tage).

Explorer Icon und Label

USBDLM kann ein paar Werte in die Windows Registry schreiben, damit der Windows Explorer für ein Laufwerk ein spezielles Icon und Bezeichnung zeigt (DefaultIcon und DefaultLabel).

Allerdings funktioniert das nicht immer - manchmal geht's, manchmal nicht. An USBDLM liegt's nicht. Planmäßig hat unter XP offenbar die Datenträgerbezeichnung (volume label) die höchste Priorität, dann die Registry-Einträge und zuletzt der generische Name wie "Wechseldatenträger".

Die Konfiguration ist analog zu AutoRun: Man kann Kriterien definieren, der erste Abschnitt bei dem die Kriterien passen, wird benutzt.

Beispiel für ein Kamera-Icon für Laufwerke mit dem Volume-Label ("Datenträgerbezeichnung") CANON DC:

[Explorer]
VolumeLabel1=CANON_DC
DefaultIcon=shell32.dll,139

Das Icon befindet sich in der XP shell32.dll bei Index 139 (gezählt ab Null).

Oder aus einer ICO-Datei, z.B. auf dem Laufwerk selbst:

; usb.ico auf dem Laufwerk nutzen falls vorhanden [Explorer]
FileExists=%drive%\usb.ico
DefaultIcon=%drive%\usb.ico

Ein DefaultLabel ("Standard-Datenträgerbezeichnung") wird normalerweise nur benutzt, wenn das Laufwerk kein eigenes Volume-Label hat. Manchmal aber doch, das Verhalten von Windows scheint hier nicht vorhersagbar.

Bei den Laufwerksbuchstaben A: und B: scheint XP besonders an seinen Floppy-Icons und Floppy-Namen zu hängen - hier ist die Warscheinlichkeit besonders hoch, dass XP die Einstellungen ignoriert.

Beispiel für eine Standard-Datenträgerbezeichnung "USB-Laufwerk" für USB-Laufwerke:

[Explorer]
BusType=USB
DefaultLabel=USB-Laufwerk

USBDLM unterstützt hier eine Reihe von Variablen.

Beispiel für den Geräte-Namen für USB-Laufwerke ohne eigene Datenträgerbezeichnung:

[Explorer]
VolumeLabel=BusType=USB
DefaultLabel=%DevName%

Wenn man ein leeres DefaultLabel haben möchte, konfiguriert man DefaultLabel=-

USBDLM schreibt dann ein Leerzeichen als DefaultLabel, da ein wirklich leerer Wert nur zur Anzeige der Standard-Bezeichnung (z.B. Wechseldatenträger) führt.

Unter Vista und Windows 7 mit dem Font "Segoe UI" kann man auch das Unicode-Zeichen U+200B (zero width space) konfigurieren - die USBDLM.INI muss dann natürlich im Unicode-Format gespeichert werden. Die unter XP üblichen Fonts Arial, Verdana usw. unterstützen dieses Zeichen leider nicht.

Wenn man kein DefaultLabel haben möchte, konfiguriert man DefaultLabel=--USBDLM schreibt dann einen leeren Wert.

Festlegen in DriveLetters-Abschnitten

Ab V4.6 können DefaultLabel und DefaultIcon gleich mit in den zuständigen DriveLetters-Abschnitt getan werden.

Wenn dort ein DefaultLabel <u>oder</u> ein DefaultIcon festgelegt sind, wird nicht nach einem passenden Explorer-Abschnitt gesucht!

Beispiel:

[DriveLetters]
DeviceID=USB\VID_9876&PID_5432
Letter=X
DefaultLabel=%DevName%

Gerät unter "Hardware sicher entfernen" zeigen oder nicht

USBDLM kann ein für ein Laufwerk einen Wert in der Registry so modifizieren, dass es unter XP nicht mehr unter "Hardware sicher entfernen" angezeigt wird. Das ist z.B. für interne Kartenleser sinnvoll, damit sie nicht versehentlich abgemeldet werden.

Beim nächsten Start des Laufwerks-Treibers wird der Wert aber zurückgesetzt, so dass er bei jedem Systemstart erneut gesetzt werden muss.

Unter Windows 7 kann "Hardware sicher entfernen" zusätzlich bei Wechseldatenträgern auch "auswerfen", so dass dort alle Wechseldatenträger aufgelistet werden. Die bekommt man unter Windows 7 also nicht ausgeblendet, aber immerhin die Funktion zum Abmelden des ganzen Gerätes deaktiviert.

Konfiguriert wird das anhand des Kriteriums "Geräte-ID", siehe auch <u>Buchstaben abh. vom der Geräte-ID</u>.

Beispiel für ein USB-Laufwerk:

[HideFromSafelyRemoveHardware] DeviceID1=USB\VID_058F&PID_6369

Oder anhand des "Friendly Name":

[HideFromSafelyRemoveHardware] DeviceID1=Kingston Flash Reader

Umgekehrt können auch Laufwerke unter "Hardware sicher entfernen" angezeigt werden, für die das normalerweise nicht der Fall ist, auch kann man obiges so rückgängig manchen:

[AddToSafelyRemoveHardware]
DeviceID1=IDE\DISKFUJITSU MHV2040BH*00000025

Aber selbst wenn man ein IDE-Laufwerk in die Liste bekommt, schlägt das Entfernen warscheinlich fehl. Mit meinem Kommandozeilen-Programm RemoveDrive kann man dagegen auch IDE-Laufwerke zum sicheren Entfernen vorbereiten.

Geräte die nichts mit Laufwerken zu tun haben, kann USBDLM auch behandeln - da USBDLM darüber aber keine Daten vorhält, muss hier die komplette Geräte-Instanz-ID angegeben werden (XP: "Geräteinstanzkennung", ab Vista: "Geräteinstanzpfad"), z.B. so:

 $[HideFromSafelyRemoveHardware] \\ CompleteDeviceID=USB\VID_1758\&PID_2004\050712014270000083 \\ CompleteDeviceID=PCI\VEN_1011\&DEV_0009\&SUBSYS_21140A00\&REV_20\3\&267A616A\&0\&50 \\ [HideFromSafelyRemoveHardware] \\ CompleteDeviceID=VID_1758\&PID_2004\050712014270000083 \\ [HideFromSafelyRemoveHardware] \\$

Die Geräte-Instanz-ID findet man im Geräte-Manager:

Das betreffende Gerät mit rechts anklicken -> Eigenschaften, Reiter "Details". Hier ist unter XP schon die "Geräteinstanzkennung" ausgewählt. Dann unten auf die Geräte-ID klicken und "Strg+C" drücken - das kopiert sie in die Windows Zwischenablage. Ein Kontextmenü mit "Kopieren" gibt's erst ab Vista.

Laufwerks-Bezeichung

Im "Hardware sicher entfernen"-Menü von XP und Vista heißen USB-Laufwerke immer "USB Massenspeichergerät" (abhängig von der Windows-Sprache) plus Laufwerksbuchstabe. USB-Laufwerke die auf NTFS-Ordner gemountet sind oder keinen Mountpunkt haben, heißen nur "USB Massenspeichergerät", was eine Unterscheidung von mehreren Laufwerken unmöglich macht. "USB Massenspeichergerät" kommt für USB-Laufwerke aber aus dem Registry-Wert "DeviceDesc" und den kann USBDLM ändern:

[Settings]

SetDeviceDesc=1

Mögliche Werte:

SetDeviceDesc=0 -> keine Änderung

SetDeviceDesc=1 -> für USB-Laufwerke ohne Laufwerksbuchstaben setzen

SetDeviceDesc=2 -> für alle USB-Laufwerke setzen

Standardmäßig schreibt USBDLM den "FriendlyName" des USB-Gerätes in den Wert "DeviceDesc", sowas wie "Corsair Flash Voyager". Benutzerdefinierte Namen mit <u>USBDLM-Variablen</u> sind auch möglich.

Beispiel für FriendlyName und BusType für alle USB-Laufwerke:

[Settings]

SetDeviceDesc=2

DeviceDesc=%FriendlyName% USB Device

Das Ergebnis wäre sowas wie "Corsair Flash Voyager USB Device".

Da USB-Geräte mehrere Laufwerke haben können (Multislot-Cardreader oder Festplattenpartitionen) ist die Nutzung von laufwerkspezifischen Variablen hier nicht sinnvoll - es würde das letzte Laufwerk "gewinnen".

Der Wert "DeviceDesc" wird auch für die Anzeige im Windows Geräte-Manager benutzt.

Unter Windows Vista haben die Original-Werte der DeviceDesc einen anderen Inhalt, z.B. diesen: @usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device

Alle Versuche, die gewünschte Bezeichnung dort "minimal invasiv" reinzuschreiben sind fehlgeschlagen, es hilft nur das komplette Ersetzen. Welche Nebenwirkungen das hat, ist mir nicht bekannt.

Bei anderen Bus-Typen als USB vergibt Windows bereits ordentliche Namen und die Bezeichnung lässt

sich wohl auch nicht ändern.

Windows 7 und neuer zeigt auch für USB-Laufwerke von Haus aus den Gerätenamen, das Ändern des DeviceDesc-Wertes wirkt sich hier nicht auf die Anzeige in "Hardware sicher entfernen" aus.

BadUSB-Geräte blockieren

Seit Jahren ist bekannt, dass USB-Geräte eine potentielle Bedrohung darstellen, da sie eine Tastatur simulieren können, die von jedem modernen Betriebssystem kommentarlos eingebunden wird und dann im Kontext des angemeldeten Nutzers Tastendrücke simulieren kann.

In 2014 hat die deutsche Firma <u>Security Research Labs</u> auf der Black Hat gezeigt, dass sich weit verbreitete, auch in USB-Speichersticks genutzte USB-Controller so umprogrammieren lassen, dass sie genau das tun - das ist eine neue Qualität des Problems.

Unter Windows lässt sich das für Tastaturen recht problemlos aushebeln, indem man die Datei C:\Windows\inf\keyboard.inf verschiebt oder die Extension ändert, so lange die Datei nicht da ist, wird keine neue Tastatur eingebunden.

Ab Vista geht's auch eleganter, siehe hier: http://www.windowspage.de/tipps/021443.html GUID_DEVCLASS_KEYBOARD: {4d36e96b-e325-11ce-bfc1-08002be10318} GUID_DEVCLASS_NET: {4D36E972-E325-11CE-BFC1-08002BE10318}

Mit USBDLM geht das etwas komfortabler: Jedes neu angeschlossene USB-Tastatur- oder USB-Netzwerkgerät wird sofort zum sicheren Entfernen vorbereitet und ist dann mit Code 21 lahmgelegt. Anschließend wird der Nutzer gefragt, ob es aktiviert werden soll. Um zu antworten braucht er natürlich eine Tastatur oder eine Maus, ggf. muss er eine USB-Maus mitbringen...

Für einen kurzen Moment funktioniert die Tastatur, aber längst nicht lange genug, um ein böses Script zu schreiben und zu starten.

Um die Rückfrage bei neu angeschlossenen USB-Tastatur-Geräten zu aktivieren:

[Settings]
BadUsbWatchKbd=1

- BadUsbWatchKbd=0 -> Neu angeschlossene USB-Tastaturen funktionieren normal (default)
- BadUsbWatchKbd=1 -> Der Nutzer wird bei einer neu angeschlossenen USB-Tastatur gefragt
- BadUsbWatchKbd=2 -> Eine neu angeschlossene USB-Tastatur wird ohne Nachfrage "sicher entfernt"

Bleibt das ultimative BadUSB-Gerät, nämlich eine USB-Tastatur die bei passender Gelegenheit böse Tastenanschläge sendet. Aber das ist schon so seit es Tastaturen gibt...

In den Raum gestellt wurden auch USB-Netzwerkadapter die DNS-Abfragen manipulieren, allerdings ohne Details wie das funktionieren soll. Auch die kann USBDLM analog behandeln:

[Settings]
BadUsbWatch**Net**=1

Titel und Text der MessageBox können konfiguriert werden. Beispiel:

[Settings]

UsbKbdMsgTitle=USBDLM

UsbKbdMsgText=USB-Tastatur angeschlossen. Aktivieren?

UsbNetMsgTitle=USBDLM

UsbNetMsgText=USB-Netzwerkadapter angeschlossen. Aktivieren?

White- und Blacklisting von Geräte-IDs und USB-Ports

Um beim Anschließen bekanntermaßen gutartiger Geräte (z.B. ein USB-WLAN-Stick) nicht jedes Mal gefragt zu werden, kann man Geräte-IDs whitelisten:

[BadUsb**White**List]

DeviceID=USB\VID_057C&PID_5601

Die Geräte-ID muss die des Tastatur- bzw. Netzwerkgerätes sein, also Geräteklasse Keyboard bzw. Net. <u>UsbTreeView</u> zeigt's.

Beispiel für ein USB-Verbundgerät mit einer HID-Tastatur:

Child Device 1 : USB-HID (Human Interface Device)

Device ID : USB\VID_05FE&PID_2001&MI_00\8&1D9FC75B&0&0000

Class : HIDClass

Child Device 1 : HID-Tastatur

Device ID : **HID\VID_05FE&PID_2001&MI_00**9&2594D27D&0&0000

Class : **Keyboard**

Der letzte Teil der Geräte-ID ändert sich an jedem USB-Port wenn das Gerät keine USB-Seriennummer hat, also besser weg lassen.

Außerdem kann man USB-Ports white- und blacklisten. Z.B. könnte es sinnvoll sein, die Rückseitigen USB-Ports eines PCs von der Rückfrage zu befreien. Beispiel:

[BadUsb**White**List]

UsbPort1=1-1

UsbPort2=1-2

UsbPort3=1-3

UsbPort4=1-4

UsbPort5=1-5

UsbPort6=1-6

Oder die Front-Ports generell für Tastaturen und Netzwerkadapter sperren. Beispiel:

[BadUsb**Black**List]

UsbPort1=1-7

UsbPort2=1-8

Für gesperrte Geräte/Ports gibt es keine Rückfrage, die Geräte werden zum sicheren Entfernen vorbereitet. Die Blacklist hat Vorrang vor der Whitelist.

Um die Portnamen zu bestimmen muss an jeden Port ein USB-Laufwerk anschließen und mit UsbDriveInfo oder <u>UsbTreeView</u> den Namen ermitteln.

Bei Ports von USB2-Roothubs ist zu beachten, dass diese meist separate Roothubs für USB1-Geräte haben, der Portname variiert also, je nachdem, ob ein USB1 oder USB2-Gerät angeschlossen wird.

Das erste Digit kann sich auch ändern, wenn ein USB-Controller hinzugefügt oder entfernt wird. Falls das in Betracht kommt, kann man die lange Notation nutzen, siehe <u>Buchstaben abhängig vom USB-Port</u>.

Gut genug?

Nicht wirklich. Eine Tastatur zu simulieren ist nur der einfachste der denkbaren Angriffe per USB, siehe unten, der letzte Link.

Links:

Heise Newsticker: BadUSB: Wenn USB-Geräte böse werden

c't uplink: BadUSB

Präsentation auf Black Hat 2005 - Barrall-Dewey

Gerät komplett ignorieren

Geräte mit merkwürdigem Verhalten kommen immer wieder mal vor. USBDLM kann bestimmte Geräte komplett in Ruhe lassen.

Konfiguriert wird das anhand des Kriteriums "Geräte-ID", siehe auch Buchstaben abh. vom der Geräte-ID.

Beispiel für den Apple iPod, dessen iTunes-Software auf den Laufwerksbuchstaben schaut und nicht gut reagiert, wenn dieser unmittelbar nach dem Anschließen geändert wird:

[IgnoreDevices]
DeviceID=ven_apple&prod_ipod
DeviceID=

Das Ignorieren des Apple iPod ist bereits fest eingebaut. Soll er dennoch von USBDLM behandelt werden, kann das über einen Abschnitt mit dem Kriterium DeviceID=ven_apple&prod_ipod erreicht werden:

[DriveLetters]
DeviceID=ven_apple&prod_ipod
Letter=X

Systemwiederherstellung

Die XP Systemwiederherstellung fühlt sich auch für externe Festplatten zuständig. Das ist völlig unsinnig, aber Microsoft war nicht willens, hier wenigstens USB- und Firewire-Platten auszunehmen, warscheinlich damit es ein paar wenige Gründe gibt, auf Vista umzusteigen - hier wird standardmäßig nur das System-Laufwerk überwacht.

Die Systemwiederherstellung kann man zwar für einzelne Laufwerke deaktivieren, das vergisst XP aber sobald ein neuer Systemwiederherstellungspunkt angelegt wird und das betreffende Laufwerk gerade nicht angeschlossen ist.

USBDLM kann versuchen, gleich beim Anschließen eines Laufwerks die Systemwiederherstellung dafür zu deaktivieren

Aber so wie es beim manuellen Deaktivieren Probleme gibt (Laufwerk fehlt in der Liste oder steht auf "offline"), klappt es auch automatisch nicht immer.

[Settings]
; nur unter XP
DisableSystemRestoreForNewDrives=1

USBDLM kann auch noch versuchen, den Ordner "System Volume Information" zu löschen:

[Settings]
DeleteSviFolder=1

Vorsicht: Falls das betreffende Laufwerk an einem anderen XP-Rechner angeschlossen war als dieser einen Systemwiederherstellungpunkt erstellt hat, und die Systemwiederherstellung war für dieses Laufwerk nicht deaktiviert, sind die zugehörigen Daten natürlich verloren und der Systemwiederherstellungpunkt ebenso unbrauchbar wie wenn das Laufwerk nicht angeschlossen ist!

Falls das Löschen fehlschlägt, kann USBDLM den Systemwiederherstellungs-Dienst kurz anhalten:

[Settings]
DeleteSviFolder=2

Um zumindest zu verhindern, dass die Systemwiederherstellung auf externen Laufwerken herumscannt und den Ordner "System Volume Information" füllt, kann man für externe Laufwerke je einen Wert in die Registry schreiben, der ursprünglich für das NT-Backup-Programm gedacht war, unter XP aber auch für die Systemwiederherstellung genutzt wird:

[Settings]

AddNewDrivesToFilesNotToBackup=1

So legt USBDLM für jedes USB-, FireWire-, TrueCrypt- und HotPlug-Laufwerk sowie für jeden Wechseldatenträger der in einen NTFS-Ordner gemountet wurde, einen Eintrag an, der dieses Laufwerk von

der Überwachung ausnimmt, s. hier:

http://www.tweakxp.com/article37472.aspx
 Die Einträge werden so angelegt, dass sie einen Windows-Neustart nicht überdauern ("volatile"). USBDLM legt sie beim Start des Dienstes ggf. wieder an.

"Scan and Fix" (überprüfen und reparieren)

Nach dem Anschließen eines FAT-formatierten USB-Wechseldatenträgers erscheint seit Windows Vista oft der "Scan and Fix"-Dialog - auf deutsch "überprüfen und reparieren".

Nachdem einmal jenes "Scan and Fix" durchgeführt wurde (alternativ ein CHKDSK /F), sollten es beim nächsten Anschließen nicht wieder kommen.

Tatsächlich kommt es oft immer und immer wieder, um dann doch keine Fehler zu finden. Es nervt einfach nur...

Warum?

Ein Blick auf den Bootsektor vor und nach dem "Scan and Fix" verrät, dass immer dann danach gefragt wird, wenn ein bestimmtes Bit gesetzt ist. Setzt man es manuell zurück, kommt keine Frage nach "Scan und Fix" mehr.

USBDLM kann genau das beim Anschließen eine USB-Laufwerkes tun:

[Settings]
PreventVistaScanAndFix=1

Was genau macht USBDLM?

Besagtes Bit im Bootsektor befindet sich im Wert "BS_Reserved1", siehe Microsofts <u>FAT32 File System Specification</u> (fatgen103.doc).

Normalerweise ist der Wert 0, ist er 1 springt "Scan and Fix" an.

Wird ein neues USB-Laufwerk angeschlossen, sucht USBDLM darauf nach einer FAT-Partition und setzt in deren Bootsektor den Wert ggf. auf 0 zurück. Gelingt das, bevor Windows das Bit liest, kommt kein "Scan and Fix". Meist klappt das rechtzeitig, aber nicht immer.

Das doktort natürlich nur am Symptom herum, besser ist es, erst gar kein gesetztes "Dirty-Bit" zu haben:

Das Bit wird immer dann gesetzt, wenn eine Dateigröße geändert sowie wenn eine Datei angelegt oder gelöscht wird.

Zurückgesetzt wird es unter Windows 7 ca. 1,5 Sekunden nach einer Größenänderung, aber erst ca. 30 Sekunden nachdem eine Datei gelöscht oder neu angelegt wurde! Ob ein Schreibcache aktiv ist, spielt offenbar keine Rolle. Das sind nur erste Beobachtungen mit meinem FAT-Dirty-Bit-Tool: WatchFatDirtyBit.zip

Also immer schön "Auswerfen" oder "Hardware sicher entfernen" nutzen, das setzt das Bit zurück auf Null.

Auch bei NTFS-Laufwerken kann "Scan and Fix" anspringen, hier kann USBDLM aber nicht helfen.

Technische Infos

Hier ein paar Hintergrundinformationen - was mir gerade so eingefallen ist. Manches findet sich auch an anderer Stelle in dieser Hilfe.

Windows-Dienst

USBDLM ist ein Windows-Dienst. Windows-Dienste werden als solche in die Registry eingetragen und vom "Service Control Manager" gestartet und angehalten.

Dienste laufen meist im Kontext "LocalSystem" und habe dort weitgehende Privilegien. USBDLM ist ein Win32-Dienst, also kein Kernel-Dienst. Als Kernel-Dienst könnte man ihn mit dem Start-Typ "LocalSystem" versehen - so wäre sichergestellt, dass er vor dem Nutzer-Login fertig geladen ist. Win32-Diensten ist das nicht vergönnt - hier ist "Automatisch" der frühestmögliche Start. Damit ist es möglich, dass der Nutzer eingeloggt wird, bevor USBDLM startet, siehe auch "USBDLM-Dienst startet zu spät" unter Probleme.

Balloontips

Unter Windows 2000 konnte man als Dienst noch problemlos Fenster auf dem Desktop des Nutzers anzeigen - alles was man ein Dienst brauchte war das "interactive" Flag.

Unter Windows XP klappt das nur für den ersten angemeldeten Nutzer, denn nur dieser läuft in der selben Windows-Session wie die Dienste.

Seit Vista ist Microsoft konsequent: Auch der erste Nutzer arbeitet in einer eigenen Session und Dienste mit dem "interactive" Flag werden mit einer Warnung im System-Log bestraft.

Unter Windows XP installiert sich USBDLM mit "interactive" Flag. Läuft der Nutzer in Session0 werden die Balloontips direkt durch den USBDLM-Dienst angezeigt, sonst startet USBDLM sich für einen Balloon-Tip selbst nochmal im Kontext des aktiven Nutzers.

Wer im direkten Anzeigen des Balloontips ein Sicherheitsrisiko sieht, kann in der Computer-Verwaltung unter Dienste bei USBDLM die Option "Datenaustausch zwischen Dienst und Desktop zulassen" deaktivieren.

Die USBLDM.MSI hat seit V4.6 kein "interacitve" Flag mehr.

Wenn USBDLM sich für einen Balloon-Tip selbst nochmal im Kontext des aktiven Nutzers startet, übergibt es die Daten für den Balloon-Tip per Kommandozeile.

Das kann man auch für eigene Balloontips nutzen, z.B. für einen Balloontip wenn ein BitLocker-Laufwerk angeschlossen wird, das Betriebssystem das aber nicht verarbeiten kann:

[OnArrival]

DeviceType=BitLocker

open="%usbdlmpath%\usbdlm.exe" -balloon -time=20000 -title="%FriendlyName%" -text1="Auf BitLocker-verschlüsseltes Laufwerk kann nicht zugegriffen werden" -icon=301

Die Balloontips sind keine Windows-Balloontips sondern handgemacht. Hauptgrund ist die begrenzte Textlänge der Windows-Balloons unter Windows XP, was z.B. bei mehreren NTFS-Mountpunkten ein Problem ist. Außerdem konnte man bis XP noch kein eigenes Icon festlegen, man braucht kein Tray-Icon und die Taskleiste poppt beim Anzeigen nicht hoch.

Die verfügbaren Icon-Nummer sind:

110 ein Laufwerk

112 zwei Laufwerke

113 ein CDROM-Laufwerk

114 ein USB-Laufwerk

115 ein Netz-Laufwerk

116 ein entferntes Netz-Laufwerk

117 eine SD-Karte

300 ein Info-Zeichen 301 ein Ausrufezeichen

Message-Boxen

Die USBDLM usr.exe kann eine Message-Box anzeigen, z.B. so:

[On**Unsafe**Removal]

open="%USBDLMPATH%\USBDLM_usr.exe" -msgbox -type:0 -icon:0x30 -title:Warnung -text:%BusType%-Gerät %FriendlyName% wurde unsicher entfernt. Schäm Dich!

Type, icon und Rückgabewert (Errorlevel) entsprechen denen der Windows-API-Funtion <u>MessageBox</u>. Type bezieht sich auf die angezeigten Schaltflächen wie z.B. MB_OK (0..6), icon auf das anzuzeigende Symbol wie z.B. MB_ICONERROR (0x10, 0x20, 0x30 oder 0x40).

TrueCrypt, Netz- und Subst-Laufwerke

TrueCrypt-Volumes sind keine "vollwertigen" Volumes, so kommt z.B. beim Mounten eines TrueCrypt-Volumes keine Benachrichtigung wie bei "richtigen" Laufwerken. Ebenso verhält es sich mit Netz- und Subst-Laufwerken.

Was bleibt ist eine schon bei Windows 3.0 vorhandene Benachrichtigung über neue Laufwerke bzw. eingelegte CDs, die auch bei Windows 7/8 noch an alle Fenster verschickt wird.

USBDLM hat da als Dienst schlechte Karten, wenn der Nutzer in einer anderen Session läuft, was seit Vista immer der Fall ist. Unter XP läuft der Nutzer in einer anderen Session wenn er nicht der erste gleichzeitig lokal angemeldete ist.

Hier braucht es ein Hilfsprogramm, das im Kontext des Nutzers mit einem unsichtbaren Fenster läuft und eben jene Benachrichtigung "nach unten" an den Dienst meldet. Genau das ist die USBDLM_usr.exe. Die wird geladen, wenn

- ExternalMountEvents=1
- es gibt ein [OnArrival] mit DeviceType=TrueCryptVolume
- es gibt ein [OnArrival] mit DeviceType=SubstDrive
- es gibt ein [OnArrival] mit BusType=remote
- [BalloonTips] ShowTrueCrypt=1
- [BalloonTips] ShowVirtual=1
- USBDLM Win32 unter Vista/Win7 x64 mit NoMediaNoLetter=1

Mit dem letzten Punkt wird <u>ein Windows-Bug</u> in der WoW64-Schicht umschifft, der wohl nicht mehr behoben wird - Windows 7 hat ihn auch noch. Mit der x64-Version von USBDLM hat sich das aber erledigt.

Seit Vista kommt noch die User Account Control (UAC) ins Spiel - hier ist der Nutzer-Kontext aufgeteilt in einen eingeschränkten und einen nicht eingeschränkten. Um Benachrichtigungen für beide zu empfangen, braucht es die USBDLM_usr zweimal. Die zweite Instanz wird geladen, wenn

- ein [OnArrival] mit DeviceType=SubstDrive mit elevated=1
- ein [OnArrival] mit BusType=remote mit elevated=1
- [BalloonTips] ShowTrueCrypt=2 oder ShowVirtual=2

Die Automatik kann man auch überstimmen:

[Settings]
LoadUsbdlm_Usr=0

- 0: nicht laden
- 1: laden wie benötigt (default)
- 2: immer laden (bei aktiver UAC für den eingeschränkten Nutzer)
- 3: immer laden (bei aktiver UAC für den eingeschränkten und den 'elevated' Nutzer)

Ereignisanzeige

USBDLM kann beim Dienst-Start/Stop, sowie beim Anschließen und Entfernen von Laufwerken und Ausführen von Autoruns Einträge in die Windows Ereignis-Protokollierung, das Event Log, schreiben:

[Settings]

EventLog=1

Es werden dann solche Einträge in der Rubrik "Anwendung" geschrieben:

USBDLM service started

Drive attached: Name='FUJITSU MHK2120AT USB Device' Type=FIXED MountPoints='Z:\'

User=admin

Drive removed: Name='FUJITSU MHK2120AT USB Device' Type=FIXED MountPoints='Z:\'

User=admin

Open executed: MountPoint=Z:\ CmdLine=C:\Windows\Systen32\cmd.exe /c C:\Backup1.cmd Z:\

USBDLM service shut down

Ab V4.3 kann das mittels <u>USBDLM-Variablen</u> frei konfiguriert werden:.

Beispiel:

[Settings]

EventLogStringArrival=Drive attached: Name='%DevName%' Type=%DriveType%

MountPoint=%Root% User=%UserName%

EventLogStringRemoval=Drive removed: Name='%DevName%' Type=%DriveType%

MountPoint=%Root% User=%UserName%

EventLogStringOpen=Open executed: MountPoint=%Root% CmdLine=%CmdLine%

EventLogStringSvcStart=USBDLM service started

EventLogStringSvcStop=USBDLM service stopped

EventLogStringSvcShutdown=USBDLM service shut down

Fall ein Eintrag nicht gewollt ist, kann man ein Minus konfigurieren, z.B.: EventLogStringOpen=-

Hinweis: Während die Windows Ereignisanzeige (der EventViewer) geöffnet ist, kann die USBDLM.EXE weder gelöscht noch ersetzt werden. Umbenennen ist aber möglich.

Ab V5.2 kann USBDLM diese Zeilen auch mit LogLevel 0 in seine Logdatei schreiben:

[Settings]

EventLog=2

Falls beide Ausgabeziele gewünscht sind: EventLog=3

TrueCrypt / VeraCrypt

TrueCrypt ist eine bewährte Open-Source-Software zur Datenträger-Verschlüsselung. Die verschlüsselten Daten werden entweder in einer Datei, einer Partition oder einem ganzen Laufwerk gespeichert.

Leider hatten die Entwickler der Software keine Lust mehr, die Software weiter zu pflegen und haben sie kurzerhand für potentiell unsicher erklärt, weil sie Fehler enthalten könnte. Ein zur selben Zeit laufendes externes Review der Quelltexte hat aber keine nennenswerten Problem zu Tage gefördert, der letzten funktionsfähigen Version 7.1a kann somit wohl weiter vertraut werden: http://www.heise.de/download/truecrypt.html

Auf Basis der TrueCrypt-Quelltexte wurde das VeraCrypt-Projekt ins Leben gerufen, Funktionalität, Bedienung und Kommandozeilenparameter entsprechen weitgehend denen von TrueCrypt, nur das Mounten von Volumes ist etwas langsam, um Bruteforce-Angriffe zu erschweren. Seit man beim Mounten den Hashalgorithmus vorwählen kann, geht es hinreichend schnell. https://veracrypt.codeplex.com

Wenn im Folgenden von TrueCrypt die Rede ist, gilt das so auch für VeraCrypt.

Ein TrueCrypt-Container wird mit Zufallsdaten initialisiert - die verschlüsselten Daten sehen dann ebenso zufällig aus. Wird eine ganze Partition als TrueCrypt-Container benutzt, hat diese kein Dateisystem und damit auch keine Datenträgerbezeichnung, bekommt von Windows aber dennoch einen Laufwerksbuchstaben. Als Dateisystem zeigt Windows dann vielsagend "RAW".

Nur die TrueCrypt-Anwendung mit ihrem Treiber kann den Container als <u>zusätzliches</u> Laufwerk mit den entschlüsselten Daten einbinden.

Datei als Container

USBDLM kann beim Anschließen eines Laufwerks nach dem Vorhandensein einer Datei schauen und diese dann durch TrueCrypt mounten lassen:

[OnArrival]

FileExists=%drive%\geheim.tc open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\geheim.tc" /letter J

Ab besten nur dann, wenn der Zielbuchstabe auch frei ist - sonst sagt TrueCrypt "Volume is already mounted":

[OnArrival]

FileExists=%drive%\geheim.tc
DirNotExists=J:\
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v "%drive%\geheim.tc" /letter J

Andererseits ist eine Fehlermeldung gar nicht schlecht, ohne wird der TrueCrypt-Container einfach nicht eingebunden.

Partition als Container

Den Laufwerksbuchstaben einer als TrueCrypt-Container benutzen Partition sollte man entfernen - er ist nutzlos und birgt das Risiko, dass das Laufwerk fix mal formatiert wird... Das kann man in der Windows Datenträger-Verwaltung erledigen oder USBDLM überlassen.

USBDLM kann den Buchstaben jeder als TrueCrypt-Container genutzten Partition entfernen. Die Erkennung eines TrueCrypt-Containers ist simpel gestickt: Wenn die Daten im ersten Sektor eine Volumes nach Zufallsdaten aussehen gibt USBDLM dem Volume den DeviceType TrueCrypt. Wenn der Container bereits durch TrueCrypt gemountet ist, kann USBDLM keine Test-Daten lesen, weil TrueCrypt sich exklusiven Zugriff gönnt. Der Lese-Versuch wird mit dem Fehlercode ERROR_SHARING_VIOLATION verweigert. USBDLM vergibt dann den DeviceType ReadSharingViolation. Das passiert typischerweise, wenn USBDLM neu gestartet wird und somit das TrueCrypt-Volume beim Test bereits gemountet ist.

Die typische Behandlung von TrueCrypt-Partitionen sieht so aus: Laufwerksbuchstaben entfernen und TrueCrypt zum Mounten starten.

Wenn der USBDLM-Dienst neu gestartet wird, während eine TrueCrypt-Partition gemountet ist, soll zwar der Laufwerksbuchstaben entfernt werden bzw. entfernt bleiben, TrueCrypt aber nicht erneut aufgerufen werden. Genau das macht das folgende Beispiel:

; Buchstabe entfernen, da nutzlos [DriveLetters] DeviceType1=TrueCrypt DeviceType2=ReadSharingViolation Letter=-

; Container auf J: mounten
[OnArrival]
DeviceType=TrueCrypt
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v %PartitionName% /letter J

; und ein Explorer-Fenster mit dem gemounteten TrueCrypt-Laufwerk öffnen [OnArrival]
DeviceType=TrueCryptVolume
open="%windir%\explorer.exe" %root%

%PartitionName% ist eine USBDLM-<u>Variable</u> die USBDLM durch sowas wie \Device\Harddisk2\Partition1 ersetzt, so wie TrueCrypt das erwartet.

Für VeraCrypt ist der DeviceType des gemouteten Volumes VeraCryptVolume. Der DeviceType eines Partitions-Containers ist aber auch bei VeraCrypt TrueCrypt, da es immer nur Zufallsdaten sind, die sich äußerlich nicht unterscheiden und TrueCrypt war zuerst da...

Unbekannte Partition als Container

Um von vornherein zu verhindern, dass Windows einer TrueCrypt-Container-Partition einen Laufwerksbuchstaben zuordnet, kann man der Partition einen unbekannten Partitionstyp verpassen, z.B. 0x7C. Es gibt dann trotzdem ein Kernel-Objekt wie \Device\Harddisk2\Partition1, Windows erzeugt aber kein Standard-Volume, somit hat man auch keinen Laufwerksbuchstaben. Allerdings wird dann auch USBDLMs OnArrival nicht ausgeführt - es gibt ja kein Volume. Hier kann man ab USBDLM V4.8 auf OnDiskArrival usw. zurückgreifen, siehe unten, "Ganzes Laufwerk als Container".

Das Ändern des Partitions-Typs geht z.B. mit dem <u>Active@ Partition Manager 3.0</u>: Rechtsklick auf das Laufwerk, Partition Table..., hier den Wert "File system [hex]" auf 7C setzten.

Ganzes Laufwerk als Container

Im ersten Sektor wird eigentlich ein Partitionstabelle erwartet. Die Windows Datenträgerverwaltung meint dann auch, ein TrueCrypt-Laufwerk "initialisieren" zu müssen, was dem Container sicher nicht gut bekommt. Schlimmer als die Empfehlung des Windows-Explorers eine TrueCrypt-Partition zu formatieren ist das aber auch nicht.

Ab V4.8 gibt es die Ereignisse OnDiskArrival, OnDiskRemovalRequest und OnDiskRemoval, so dass ganze TrueCrypt-Laufwerke behandelt werden können.

Die nötige Disk-Nummer ist bekannt, sie ist identisch mit der DeviceNumber. Partition0 steht unter Windows für die ganze Disk:

; Disk auf J: mounten
[On**Disk**Arrival]
DeviceType=TrueCrypt
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /v \Device\Harddisk**%DeviceNumber%**\Partition**0**/letter J

; J: beim "Sicher entfernen" der Disk dismounten [On**Disk**RemovalRequest]
DeviceTypes=TrueCrypt,ReadSharingViolation
open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dJ

Beim Entfernen kann der DeviceType ReadSharingViolation oder TrueCrypt sein, abhängig davon, ob das Volume beim Start des USBDLM-Dienstes schon gemountet war oder nicht.

Das TrueCrypt-Volume

Wenn TrueCrypt einen Container mountet, dann macht es dessen Inhalt unter einem neuen Laufwerksbuchstaben verfügbar. Dabei handelt es sich nicht um ein "vollwertiges" Laufwerk - einige Interna wurden entweder eingespart oder es ging nicht anders.

USBDLM sieht die Ankunft des neuen Laufwerksbuchstaben und kann dann einen OnArrival auslösen. Der DeviceType des gemounteten TrueCrypt-Laufwerks ist TrueCryptVolume, der eines VeraCrypt-Laufwerks ist VeraCryptVolume.

Mehr kann USBDLM mit einem TrueCrypt-Volume nicht machen. Es kennt dessen Laufwerkbuchstaben und macht ihn auch in den USBDLM-Variablen wie %drive% und %root% verfügbar. Das Ändern des Laufwerksbuchstaben wäre zwar möglich aber dann verliert TrueCrypt den Überblick - deshalb unterstützt USBDLM das nicht.

Das TrueCrypt-Laufwerk zusätzlich in einem (bereits vorhandenen) NTFS-Ordner bereitzustellen, kann man sich aber basteln:

[OnArrival]

DeviceType=TrueCryptVolume

open="%windir%\System32\mountvol.exe" "C:\TrueCrypt" %VolumeName% system=1

open="%windir%\explorer.exe" %root%

USBDLM weiß nichts vom Zusammenhang zwischen dem TrueCrypt-Volume und dessen Container - es arbeitet nur aufeinanderfolgende Ereignisse ab.

Sicheres Entfernen

Beim "Vorbereiten zum sicheren Entfernen" des Wirtslaufwerks kann USBDLM das TrueCrypt-Volume unmounten lassen, TrueCrypt will dafür den Laufwerksbuchstaben des TrueCrypt-Volumes sehen:

[OnRemovalRequest]

FileExists=%drive%\geheim.tc

DirExists=J:\

open="%ProgramFiles%\TrueCrypt\TrueCrypt.exe" /q /dismount J

Der Parameter /q oder /quit ist wichtig: USBDLM beantwortet die Anfrage zum "Sicheren Entfernen" erst, wenn die gestartete TrueCrypt.exe beendet wurde.

Ohne /quit zeigt TrueCrypt nach dem Dismounten ein Fenster und USBDLM wartet und wartet... Letztlich wird Windows dann sagen "Der Dienst USBDLM verhindert das Entfernen"...

Zum Abmelden eines TrueCrypt-Volumes (TrueCrypt nennt das "Dismount") habe ich ein Kommandozeilen-Tool geschrieben, dass auch den Containernamen als Laufwerksinfo akzeptiert. Außerdem kann es endlos weiter versuchen bis es klappt und auch offene Handles auf dem TrueCrypt-Volume anzeigen.

Ab USBDLM V5.0 gibt es das Kriterium IsMountedTrueCryptContainer, das zutreffend ist, wenn der angegebene Container (eine Datei, eine Partition oder eine Disk) als TrueCrypt-Laufwerk gemountet ist:

; wenn geheim.tc auf dem zu entfernenden Laufwerk von TrueCrypt gemountet ist, dieses freigeben [OnRemovalRequest]

IsMountedTrueCryptContainer=%drive%\geheim.tc open="C:\Tools\EjectTcv.exe" %drive%\geheim.tc -L -H

; wenn das zu entfernende Volume von True Crypt gemountet ist, dieses freigeben $\left[OnRemoval Request \right]$

IsMountedTrueCryptContainer=%PartitionName% open="C:\Tools\EjectTcv.exe" %PartitionName% -L -H

Mit einem gepatchten Partitions-Typ oder wenn die ganze Disk als Container genutz wird, gibt es kein Volume, somit auch kein OnRemovalRequest. Hier muß On**Disk**RemovalRequest genutzt werden, die

Partitionsnummer muß bekannt sein und Partitionsname selbst gebaut werden:

; TrueCrypt-Container in Paritition1 freigeben
[On**Disk**RemovalRequest]
IsMountedTrueCryptContainer=\Device\Harddisk**%DeviceNumber%**\Partition1
open="C:\Tools\EjectTcv.exe" \Device\Harddisk**%DeviceNumber%**\Partition1 -L -H

; TrueCrypt-Container der Disk freigeben
[On**Disk**RemovalRequest]
IsMountedTrueCryptContainer=\Device\Harddisk**%DeviceNumber%**\Partition**0**open="C:\Tools\EjectTcv.exe" \Device\Harddisk**%DeviceNumber%**\Partition**0** -L -H

EjectTrueCryptVolume

Ab V2.7 funktioniert EjectTcv auch mit VeraCrypt.

BitLocker To Go

BitLocker ist die mit Windows Vista eingeführte Laufwerksverschlüsselung. BitLocker unterstützt ausschließlich lokale, nicht entfernbare Datenträger.

Mit Windows 7 wurde "BitLocker To Go" eingeführt. Damit lassen sich Hotplug-Laufwerke verschlüsseln. Das Einrichten eines verschlüsselten Laufwerks wird nur von den Editionen Ultimate und Enterprise unterstützt, Zugriff ist aber mit allen Windows 7 Editionen möglich, mit Windows Server 2008 R2 dagegen nicht.

BitLocker funktioniert über einen Filtertreiber, der den Zugriff auf das Laufwerk zulässt oder blockiert.

"BitLocker To Go"-Laufwerke gibt es in zwei Varianten, es hängt vom Dateisystem ab, welche man bekommt:

Mit BitLocker verschlüsselte FAT/FAT32/exFAT-Laufwerke haben auch ein unverschlüsseltes FAT-Dateisystem mit einigen Dateien darauf, die man auch auf Windows ohne "BitLocker To Go" zu sehen bekommt. Die meisten Dateien tragen das Dateiattribut "versteckt", so dass die meisten Nutzer sie nicht sehen. Nicht versteckt ist die Datei BitLockerToGo.exe, das ist das "BitLocker To Go-Lesetool". Über die ebenfalls versteckte autorun.inf soll die BitLockerToGo.exe gestartet werden. Falls das nicht funktioniert, kann man sie manuell starten. Die BitLockerToGo.exe öffnet ein Explorer-artiges Fenster, das einem Lesezugriff auf das verschlüsselte Laufwerk verschafft.

BitLocker-Laufwerke die mit NTFS formatiert sind, bieten dieses Feature nicht. Windows ohne Unterstützung für BitLocker To Go schlägt dann vor, das Laufwerk zu formatieren.

Was USBDLM tun kann

1) Auf Windows-Editionen ohne Unterstützung für "BitLocker To Go"

Da FAT-formatierte BitLocker-Laufwerke ein gültiges Dateisystem haben, bekommen sie keine spezielle Behandlung. USBDLM kann hier per OnArrival-Abschnitt die BitLockerToGo.exe starten falls vorhanden und sonst den Laufwerksbuchstaben entfernen:

Auf **FAT**-Laufwerken braucht's keinen DeviceType, man schaut nach der BitLockerToGo.exe und startet sie:

[OnArrival]

FileExists=%drive%\BitLockerToGo.exe open=%drive%\BitLockerToGo.exe

USBDLM erkennt **NTFS**-BitLocker-Laufwerke als solche - sie bekommen die DeviceTypes BitLocker und UnrecognizedVolume.

Mit NTFS-formatierten Bitlocker-Laufwerken kann man eh nichts anfangen - am besten den Laufwerksbuchstaben wegnehmen damit Windows nicht vorschlägt, es zu formatieren:

[Driveletters]
DeviceType=BitLocker
Letter=-

Außerdem könnte man einen handgemachten BalloonTip anzeigen, der über das BitLocker-Problem informiert:

[OnArrival]

DeviceType=BitLocker

open="%usbdlmpath%\usbdlm.exe" -balloon -time=20000 -title="%FriendlyName%" -text1="Auf BitLocker-verschlüsseltes Laufwerk kann nicht zugegriffen werden" -icon=131

2) Auf Windows-Editionen mit Unterstützung für "BitLocker To Go"

Hier werden alle "BitLocker To Go"-Laufwerke als solche erkannt. AutoRun-Ereignisse werden dann erst für das entschlüsselte Laufwerk getriggert. Kriterien wie VolumeLabel, VolumeSerial oder das Vorhandensein einer Datei funktionieren erst wenn das Laufwerk entsperrt ist, für das Zuordnen von Laufwerksbuchstaben ist es dann aber schon zu spät, es funktionieren also nur Kriterien, die vor dem Entsperren ermittelt werden können.

Zunächst setzt USBDLM den DeviceType BitLocker, sobald das Laufwerk entsperrt wurde, wird der DeviceType BitLocker entfernt.

Probleme

Falscher Laufwerksbuchstabe für einen kurzen Moment

Ein neu angeschlossenes Laufwerk bekommt von Windows den zuletzt zugeordneten oder den ersten freien Buchstaben. USBDLM ändert diesen dann ggf. zum frühestmöglichen Zeitpunkt.

Wenn aber eine Anwendung auch diesen frühestmöglichen Zeitpunkt nutzt, sieht sie u.U. den alten Buchstaben, den USBDLM dann gleich ändert.

Um das zu vermeiden, kann USBDLM nach dem Entfernen eines Laufwerks einen "No-Letter"-Wert in die Registry schreiben durch den das Laufwerk beim nächsten Mal keinen Buchstaben bekommt - USBDLM ordnet dann einen zu.

[Settings]

DeleteLettersOnRemoval=1

Normalerweise ist das aber nicht nötig.

Ab V4.8.6 USBDLM schreibt USBDLM bei Ankunft des Disk-Devices einen No-Letter-Eintrag in die Window-Mountmanager-Registrydatenbank, so dass das Volume zunächst keinen Buchstaben bekommt und USBDLM direkt den gewünschten zuordnen kann. Bis einschließlich Windows 7 funktioniert das gut, Windows 8 ist zu schnell. Zumindest kann USBDLM den falschen Buchstaben etwas früher löschen als zuvor.

USBDLM-Dienst startet zu spät

Seit XP wartet Windows mit dem Anzeigen des Login-Dialogs (oder mit dem automatischen Login) nicht mehr, bis alle Dienste gestartet sind - die Illusion vom schnellen Systemstart hat Vorrang. So kann es insbesondere bei Nutzung eines automatischen Logins passieren, dass USBDLM beim Rechner-Start bereits angeschlossene USB-Laufwerke nicht rechtzeitig ummounten kann, um sie z.B. für Netzlaufwerke frei zu machen.

Ab V4.5.1 kann USBDLM das Ausführen von anderen AutoRuns und das Laden des Explorers verzögern bis der USBDLM-Dienst gestartet ist:

[Settings]

DelayDesktop=1

Das funktioniert ganz simpel: Beim Herunterfahren erzeugt USBDLM einen RunOnce-Eintrag in der Registry, so dass Windows beim Hochfahren USBDLM mit dem Parameter -wait4usbdlmsvc als unsichtbares Programm startet und auf dessen Beendigung wartet. Diese USBDLM-Instanz wiederum wartet so lange bis der USBDLM-Dienst läuft und beendet sich dann und Windows setzt den Startvorgang fort

Da Windows mit dem Ausführen weiterer AutoRuns und mit dem Laden des Deskops wartet bis alle RunOnce-Einträge abgearbeitet sind, hat USBDLM und auch andere Dienste Zeit zum rechtzeitigen Laden. Insgesamt geht kaum Zeit verloren, nur die Illusion vom schnellen Start wird verdorben...

Löschen von Verzeichnissen schlägt fehl, wenn Laufwerk in NTFS-Ordner gemountet

Da kann USBDLM nichts dafür, es ist eines von vielen Problemen des Windows Explorers, wenn NTFS-Ordner als Mountpunkte genutzt werden, siehe "Mounten in NTFS-Ordner" unter Konfiguration.

Laufwerksbuchstaben kehren zurück

Wenn Laufwerksbuchstaben die per NoMediaNoLetter=1 entfernt wurden oder wenn Laufwerke in NTFS-Ordner gemountet werden, kehren die Laufwerksbuchstaben manchmal überraschend zurück... Dann meint irgendeine tolle Software, alles besser zu wissen und ordnet einfach Laufwerksbuchstaben zu. Bekannt dafür ist das U3 Launchpad auf U3-Sticks (die U3launch.exe auf dem Pseudo-CDROM), siehe auch Kein Medium, kein Buchstabe.

Eine andere Software die Laufwerksbuchstaben zuordnet ist "Secure Storage Device SDK" (SSDService.exe) von MXI.

USBDLM.EXE kann nicht gelöscht oder ersetzt werden

Während die Windows Ereignisanzeige (der EventViewer) geöffnet ist, kann die USBDLM.EXE weder gelöscht noch überschrieben werden, umbenennen ist aber möglich. Das ist natürlich auch der Fall, während der USBDLM-Dienst läuft...

Geräte die nicht starten unter Vista, Windows 7, Windows 8

Für USB-Wechseldatenträger installiert Vista/Win7/8 immer auch ein "WPD Dateisystem". WPD steht für "Windows Portable Device", eine Programmierschnittstelle für Geräte wie MP3-Player, Handys u.a. Seit Windows 7 läuft das Ganze im Gerätemanager unter "Tragbare Geräte".

Wenn ein USB-Wechseldatenträger keinen Laufwerksbuchstaben hat (egal ob USBDLM den entfernt oder ob der Nutzer es manuell erledigt), schlägt der Start des Treibers mit "Code 10" fehl. Macht aber nichts: Als normaler Datenträger funktioniert das Laufwerk trotzdem.

Wenn man den WPD-Kram komplett loswerden will, kann man den Dienst "Enumeratordienst für tragbare Geräte" (WPDBusEnum) deaktivieren. Nebeneffekt: Der Windows Media Player synchronisiert sich dann nicht mehr mit USB-Massenspeichern. Und USB-Geräte die nur WPD verstehen funktionieren natürlich nicht mehr...

Ab Windows 8.1 gibt's das Problem nicht mehr.

Vista / Win7+ Virtual Store

Die mit Windows Vista eingeführte "User Account Control", kurz UAC, verweigert auch Administratoren den Schreibzugriff auf systemrelevante Verzeichnisse. Damit Software die Daten im ihrem Verzeichnis unter "C:\Programme" schreibt dennoch funktioniert, wird ihr nicht etwa der Schreibzugriff auf's eigene Verzeichnis gestattet, nein, er wird kommentarlos "umgeleitet" in den "Virtual Store". Das ist für Dateien das Verzeichnis

Users/(UserName)/AppData/Local/Virtual Store/Program Files/(Programm).

Die betreffenden Dateien gibt's dann mehrfach, einmal im Original und weitere Male für jeden Nutzer unter dessen Konto schreibend darauf zugegriffen wurde.

USBDLM bekommt als Windows-Dienst immer das Orignial zu sehen, der eingeschränkte Administrator schreibt die USBDLM.INI aber in den Virtual Store. Ich hatte darin kein Problem gesehen, weil beim Test mit den Windows-Editor "Notepad" das Speichern verweigert wurde, wohl weil die notepad.exe unter C:\Windows und nicht unter C:\Programme liegt. Fremd-Editoren die unter C:\Programme installiert sind, werden aber in den Virtual Store umgeleitet...

Ich wollte die USBDLM.INI aber unter Programme haben, eben weil sie dort nur von Admins geändert werden darf. Ab V4.3 bringt USBDLM eine Warnung wenn im Virtual Store eine USBDLM.INI gefunden wurde - beim Klick auf diesen Balloon wird ein Explorer-Fenster mit diesem Verzeichnis geöffnet.

Lösungen:

- immer daran denken, die USBDLM.INI als echter Administrator ("elevated") zu bearbeiten, ab V4.7.3 gibt's dafür die _edit-ini.cmd
- USBDLM z.B. in C:\USBDLM oder C:\Tools\USBDLM installieren der Virtual Store kümmert sich nur um C:\Programme. USBDLM setzt dann die Schreibrechte auf EXEen und INI auf "nur Admins".
- Nutzung des eigentlich für Dienste vorgesehen Daten-Verzeichnisses dort sucht USBDLM zuerst nach der INI. Das Verzeichnis ist aber unter jeder Windows-Version anderswo versteckt: Unter Vista ist das "C:\Windows\System32\config\systemprofile\AppData\Roaming\USBDLM". Unter Vista x64 wird "C:\Windows\System32" für Win32-Programme wie USBDLMs Win32-Version nach "C:\Windows\SysWOW64" umgeleitet...

Erkennung von TrueCrypt-Laufwerken

Ganze Laufwerke die als TrueCrypt-Container benutzt werden, kann USBDLM nur erkennen solange sie nicht durch TrueCrypt gemountet sind. TrueCrypt gönnt sich exklusiven Zugriff, so dass USBDLM keine Test-Daten lesen kann - der Versuch schlägt mit dem Fehler ERROR_SHARING_VIOLATION fehl. Aushilfsweise bekommt so ein Volume dann statt des DeviceTypes "TrueCrypt" den DeviceType "ReadSharingViolation". Mehr dazu hier: TrueCrypt

Fehler 193

Wenn der Start des USBDLM-Dienstes mit Fehler 193 fehlschlägt (ERROR_BAD_EXE_FORMAT), liegt das warscheinlich an unzureichenden Zugriffsrechten auf die USBDLM.EXE. Das Konto "SYSTEM" muss für den USBDLM-Ordner mindestens das Recht "Lesen, Ausführen" haben.

Einstellen kann man das mit Rechtsklick auf den Ordner -> Eigenschaften -> Reiter "Sicherheit".

Generieren einer MSI-Datei

Mit einer MSI-Datei kann man USBDLM mittels Gruppenrichtlinie im Netzwerk verteilen - Admin wissen das besser als ich...

Die USBDLM.MSI die bis einschließlich V4.5.9 geliefert wurde, ist mit OnDemand WinInstall LE generiert.

Bug: Für alle Versionen habe ich identische Product- und Package-IDs benutzt. Vorteil: Mit jeder Version kann man eine andere deinstallieren. Nachteil: Updates funktionieren nicht, für ein Update muss man deinstallieren und erneut installieren.

Ab V4.6 wird das Microsoft WIX toolset genutzt. Hier sind im WiX-Script für Product- und Package-ID ein "*" konfiguriert, so dass WiX immer neue GUIDs generiert. So funktionieren Updates, zum Deinstallieren muss aber die selbe Version wie zur Installation genutzt werden.

Zum Updaten sehr alter Versionen bitte zuerst V4.4 installieren, dann erst die aktuelle Version. Das MSI benutzt zum Anhalten des USBDLM-Dienstes dessen Kommandozeilenparameter -silentstop und diesen gibt es erst seit V4.4.

Hier gibt's das WIX-script für USBDLM: USBDLM.WXS.ZIP

Und hier das <u>Microsoft WIX</u> Toolset. Die WIX3x.MSI herunterladen, mit recht anklicken -> Installieren. Falls ein Warndialog kommt, mit OK bestätigen (da geht's nur um die Visual Studio Integration).

WiX installiert sich in den Standard-Programme-Ordner, typischerweise "C:\Programme\WiX Toolset v3.8".

WiX benötigt das .NET Framework 3.5 SP1.

Ab Windows 7 funktioniert der .NET 3.5 Installer nicht mehr, dafür wird es mit Windows mitgeliefert. Aktiviert wird es über die Systemsteuerung -> "Windows Funktionen aktivieren oder deaktivieren" oder per Kommandozeile:

DISM /Online /Enable-Feature /FeatureName:NetFx3 /ALL /NoRestart

Dann die angepasste USBDLM.INI in den USBDLM-Ordner kopieren, ebenso die USBDLM.WXS. In der USBDLM.WXS USBDLM_sample.ini in USBDLM.ini ändern.

Dann im USBDLM-Ordner eine CMD- oder BAT-Datei anlegen, die das MSI generiert, etwa so:

@echo off

set WiX_Path=%ProgramFiles%\WiX Toolset v3.8\bin

if not exist "%WiX Path%" echo !!!!!!!!!! WiX nicht gefunden !!!!!!!!!! & pause & goto ExitPoint

rem -- sicherstellen, dass der USBDLM-Ordner der aktuelle ist -- cd /D %~dp0

rem -- WiX candle und light ausführen, Pause falls Fehler --

"%WiX_Path%\candle.exe" -dProcessorArchitecture=x86 -arch x86 -out USBDLM.wixobj USBDLM.wxs if errorlevel 1 pause & goto ExitPoint

"%WiX_Path%\light.exe" -out USBDLM.msi USBDLM.wixobj if errorlevel 1 pause & goto ExitPoint

rem -- aufräumen -del USBDLM.wixobj del USBDLM.wixpdb

:ExitPoint

Für die x64-Version ist zweimal x86 in x64 zu ändern: "%WiX_Path%\candle.exe" -dProcessorArchitecture=x64 -arch x64 -out USBDLM.wixobj USBDLM.wxs

Die Datei USBDLM.wixpdb wird dort angelegt, wo auch das MSI hinkommt. Wenn man also einen alternativen Ausgabepfad angibt (-out), dann diesen auch in der Zeile mit "del USBDLM.wixpdb" angeben.