USBSecure Professional 4.4

Port Security for Windows 7, Windows 8 and Windows 10

Installation and Configuration Guide

USBSecure Professional 4.4

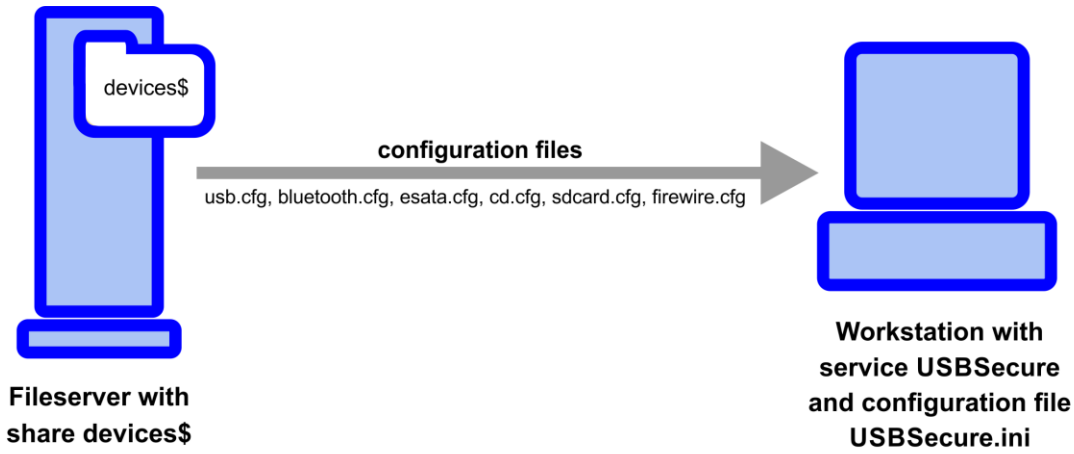
Installation Guide

Content

Functionality

USBSecure Professional is security software for securing the USB interfaces of client computers. USBSecure Professional allows you to define which user can access which USB device on a per-user basis. Additionally USBSecure Professional enables or disables Bluetooth connections, CD/DVD drives, floppy drives, FireWire ports, eSATA ports and SD card readers.

USBSecure Professional runs as a service on Windows 7, Windows 8 and Windows 10 (32 or 64 bit). You can define the access to the devices in whitelists. When a user logs on, USB devices, Bluetooth connections, CD/DVD drives, floppy drives, FireWire ports, eSATA ports and SD card readers will be enabled or disabled on the basis of the whitelists. An existing file server or any Windows server can act as a USBSecure server, only a share is required.
Additionally to the USBSecure service a scheduled task is installed during setup. This task starts a process USBSecureControl.exe to monitor the USBSecure service.



Required additional tools

USBSecure Professional 3 and 4 do not require any additional tools in contrast to version 1 and 2. All required files are included in the MSI package.

Installation

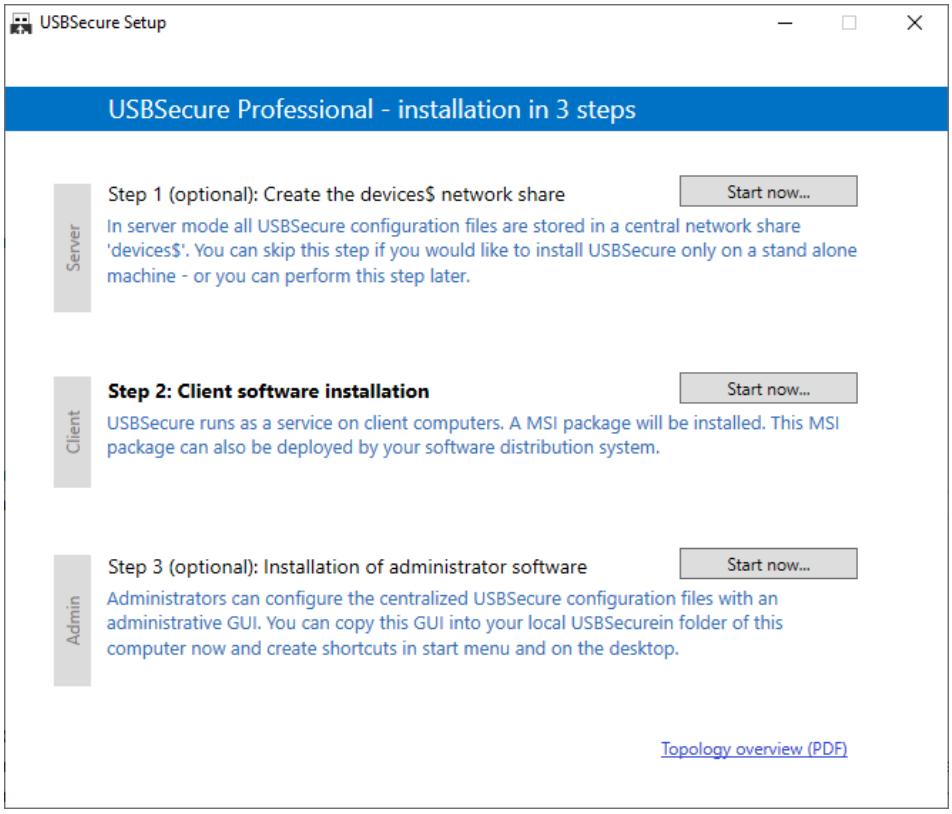
System requirements

Client: The USBSecure service (blocking / allowing devices and ports) runs without any .NET Framework software. For proper user interaction .NET-Framework 3.5 is necessary. Note: Since Windows 7 .NET-Framework 3.5 is installed by default.

Server: The USBSecure server consists only of a Windows network share with read access for everyone. Any Windows server (or client) can act as USBSecure server, also a filer or a NAS device.

Client installation

Please run Setup.exe and perform Step 2 to install the client software. USBSecure.msi will be executed. Alternatively, you can distribute the client software with you software distribution system or via Group Policies.



1. Run the file USBSecure.msi (Step 2) with administrative privileges on the client which should be protected. During installation you will be prompted for the **USBSecure destination path** and the **USBSecure server**. You can leave the name of the server blank and provide it later (USBSecure.ini) if you haven't got any USBSecure server at this time. You also need a valid license key during installation. The license key of the free 5 PC version is included in a text file in the download.
2. Note: Before you start the USBSecure service, please keep in mind that the service could disable you fingerprint sensor! If you log on via fingerprint, you should know your password.
3. Start the service "USBSecure". Logfile ***USBSecure.log*** will be generated in the USBSecure folder.

Test of the client installation

You can run USBSecure Professional now in client mode without server. Because of the entry „service = hidusb“ in file usb.cfg only keyboards and mice are allowed for all users and any USB device for user administrator (entry *). „service = usbhub“ and „service = usbhub3“ should always exist in usb.cfg.

Now connect a USB mass storage to your client, e.g. a USB stick. Because USB sticks are not allowed in usb.cfg (except for user administrator), the USB stick will be deactivated (see device manager and USBSecure.log).
Insert „service = usbstor“ in usb.cfg in the [AllUsers] section and restart the USBSecure service. Your plugged in mass storage device will be activated now and appears as a drive in Windows Explorer.

Information for Windows 7/8/10: You must start your editor in mode „Run as administrator“ to modify your usb.cfg.

Please read how to grant access to single devices and how to implement access on a per-user basis in chapter „Configuration“.

Notice: The built-in group "Users" must not have write access to the files in USBSecure folder in a productive environment.

Silent Installation

You can implement USBSecure Professional installation without any user interaction with the following command:

msiexec /i USBSecure.msi /qb USBSECURE_SERVERNAME=<Name of USBSecure server> INSTALLDIR="C:\Program Files\USBSecure" LICENSEKEY=AAAAA-BBBBB-CCCCC-DDDDD-EEEEE

Provide your license key behind the keyword **LICENSEKEY=**. You can find the license key for the free 5-PC version as a text file included in the download.

INIOVERWRITE=<time in minutes>
This value specifies when the local configuration file USBSecure.ini will be overwritten by a centralized version. The centralized USBSecure.ini must be located in the devices$ share. SBSecure.ini überschrieben wird. **<time in minutes> defines the time since the USBService start.** If you specify this MSI value it will overwrite the entry in USBSecure.ini. Example: INIOVERWRITE=15

NOUSBSTORINFO=<yes|no|warn>
Defines the behavior when inserting a forbidden USB mass storage device (see USBSecure.ini). If you specify this MSI value it will overwrite the entry in USBSecure.ini. Example: NOUSBSTORINFO=no

ADMINSCANTSTOP=<0|1>
Specifies wether the USBSecure service can be stopped / restarted by administrators or not. Value ADMINSCANTSTOP=1 prevents administrators to stop and restart the service. If you specify this MSI value it will overwrite the entry in USBSecure.ini. Example: ADMINSCANTSTOP=0

NETWORKADMINSCANSTOP=<0|1>
Only in combination with ADMINSCANTSTOP. Specifies wether the USBSecure service can be stopped / restarted by administrators over the network or not. Please note that it’s “CAN” in this case. NETWORKADMINSCANSTOP =1 allows administrators to stop and restart the USBSecure service when connected over the network, e.g. via computer management. This MSI value will NOT be inserted in USBSecure.ini. Example: NETWORKADMINSCANSTOP =1

The command for silent uninstall:

msiexec /x USBSecure.msi /qb

Server installation

Create a Windows shared folder „devices$“ on a Windows server (or a workstation which should act as a server). Grant READ permission to the group “Everyone”. Ensure that Everyone has got read permission for the folder as well as for the share. Copy files ***floppy.cfg***, ***cd.cfg, firewire.cfg, sdcard.cfg, esata.cfg*** and ***usb.cfg*** into the „devices$“ folder. You can find the files in the installation source in folder “Server” or in the client installation on your workstation in the USBSecure folder (default: C:\Program Files\USBSecure).

Notice: The USBSecure server and the client must reside in the same Windows domain.

Notice: If your devices$ share is not on a Windows server but on a storage system (filer), you might recognize that the access to your devices$ share doesn't work correctly. Especially clustered systems may not act exactly like Windows servers. In this case you should not use the name of the cluster but the real name of one node.

Test the server functionality

1. Change the value behind **Server=** in your client installation in USBSecure.ini to the hostname of your USBSecure server. If the hostname of your USBSecure server is Fileserver1, then change the line to **Server=Fileserver1**.
2. Change the line **service=USBSTOR** to **#service=USBSTOR** in your central usb.cfg file (on the USBSecure server). # indicates that the line is a comment and will be ignored.
3. Restart the USBSecure service on your client. The configuration files (usb.cfg, cd.cfg, floppy.cfg und firewire.cfg) are copied from server to client at startup of the USBSecure service. Ensure that the entry **service=USBSTOR** in you local usb.cfg is deactivated with the comment character # like in your central usb.cfg file.

All mass storage devices will now be deactivated in device manager.

If the server access doesn't work

If the access to the USBSecure server doesn't work correctly (central usb.cfg is not copied to client when restarting USBSecure service), please make sure that

- Server and client are members oft he same Windows domain
- Group **Everyone** has read permissions for the devices$ share
- Group **Everyone** has NTFS read permissions for the devices$ folder

Please check with the following command (from the command line) if the access is working with the logged on user:

```
copy \\server-name\devices$\usb.cfg
```
(replace „**server-name**" by the name of your USBSecure server)

If the copy operation succeeds you have to ensure that the USBSecure service is also able to perform the copy process. Please perform the following test:

For Windows XP

Open a command prompt on your client as the **local system account**. To achieve this, open a command prompt (as administrator) and run the following command:

```
at 10:36 /INTERACTIVE cmd.exe
```

Provide the current time plus 2 minutes instead of 10:36. If your current time is 11:17 then type 11:19. (If „access denied" appears, please log on as administrator.) At 11:19 a command prompt will appear where you can execute the following copy command:

```
copy \\server-name\devices$\usb.cfg
```
(replace „**server-name**" by the name of your USBSecure server)

If this copy operation succeeds the USBSecure server is able to perform the copy operation.

For Windows Vista / Windows 7 / Windows 8 / Windows 10

Download the Sysinternals PS Tools (http://www.microsoft.com/sysinternals) and run the following command from a command prompt (run as administrator):

```
psexec -i -s cmd.exe
```

Enter the copy command in the newly-opened command prompt:

```
copy \\server-name\devices$\usb.cfg
```
(replace „**server-name**“ by the name of your USBSecure server)

If this copy operation succeeds the USBSecure service is able to perform the copy operation.

If the copy operation fails and your devices$ share is located on a storage system, try using a real Windows server as USBSecure server. Maybe the storage system does not exactly behave like a Windows server.

If you don’t succeed with a real Windows server, please contact support@lugrain-software.com.

Service USBSecure

The USBSecure service can‘t be stopped by **Users** or **Power Users**. Only **Administrators** are able to stop and start the service. It is also possible to prevent administrators from stopping the service (see MSI values).

Uninstall

Uninstall can be performed in Control Panel / Add/Remove Software.

Note: USB devices that have been deactivated by the USBSecure service, are not activated by uninstalling the service. Enable all devices before uninstall with

[AllUsers]
*

and restart the USBSecure service or activate the devices later in device manager.

Upgrade from an older version

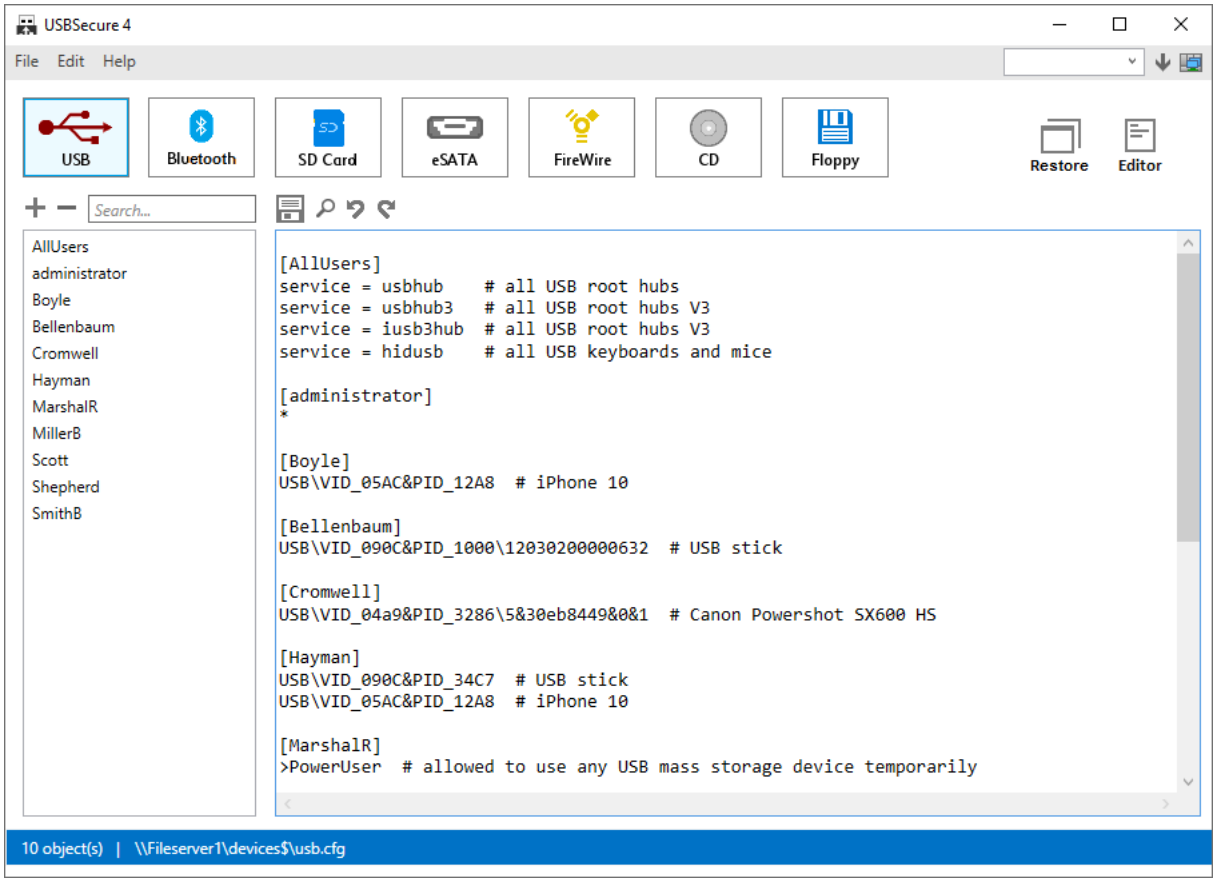
The upgrade from an older USBSecure version always consists of an uninstall and reinstall. Configuration settings are not lost in a network installation because they are stored centrally in the devices$ share. In a local installation (stand-alone) please backup all .cfg files and copy them into the folder after reinstall.

If you are upgrading from an older version to version 4.4, proceed as follows:

1. Copy the bluetooth.cfg file from the "Server" directory of the installation media to your existing devices$ share.
2. Replace the administration GUI USBSecure-Admin.exe with the new version from the "Admin" directory on the installation media.
3. Reinstall the USBSecure.msi client software from the "Client" directory of the installation media.

Configuration

From USBSecure Professional version 4.0 the .cfg files (whitelists) can be configured with a graphical user interface. The GUI requires .NET Framework 4.5 on your computer. It is also supported to configure the whitelists directly with an editor.



Configuration files

The configuration files ***floppy.cfg***, ***cd.cfg, firewire.cfg, esata.cfg, sdcard.cfg, bluetooth.cfg*** and ***usb.cfg*** are whitelists containing allowed devices. Users not listed in these files do not have access to the devices.
The [AllUsers] section in ***usb.cfg*** and the AllUsers entry in floppy.cfg, cd.cfg, esata.cfg, sdcard.cfg and firewire.cfg are valid for all users. The .cfg files are maintained centrally in the devices$ share. They are copied to the client's USBSecure folder whenever the service starts. The client solely uses the locally stored .cfg files.

USBSecure.ini is a static file in the client's USBSecure folder. It contains global settings (name of the USBSecure server, loglevel etc.).

New in USBSecure Professional version 3 / 4: parameter **IniOverwrite** for centralized management of the USBSecure.ini file (see below).

floppy.cfg, cd.cfg, esata.cfg, firewire.cfg and sdcard.cfg

Users which should have access to floppy drives, CD/DVD drives, FireWire ports, eSATA devices or SD cards, must be listed in the files ***floppy.cfg, cd.cfg, esata.cfg, sdcard.cfg*** and ***firewire.cfg*** (one user per line):

```
UserA
UserB
UserC
```

If you would like to configure all users to have access, you can make the following entry „AllUsers“ in the appropriate file:

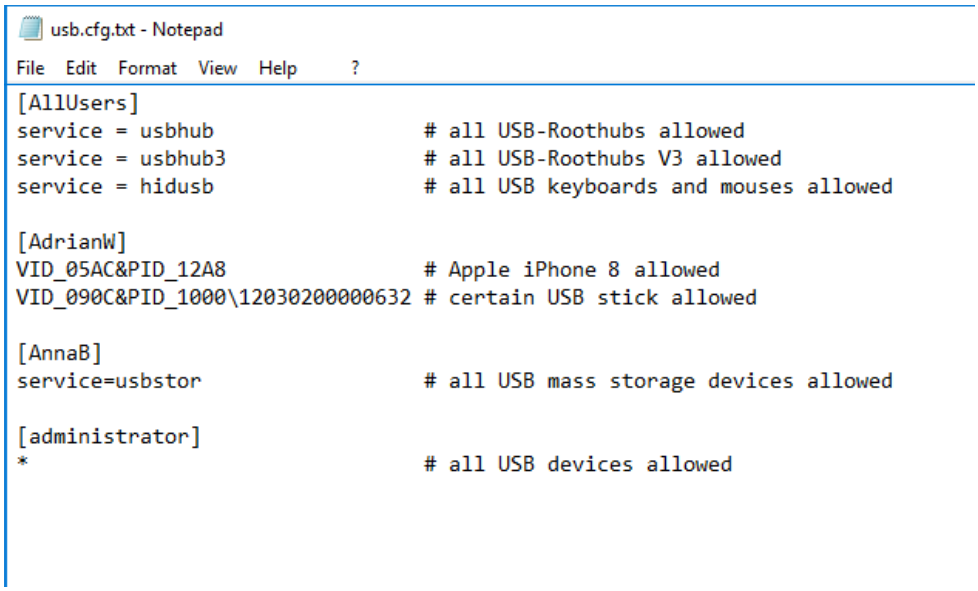
```
AllUsers
```

usb.cfg

File usb.cfg manages the user's access to USB devices. User names must be written in brackets **[ ]**. Below the user name the allowed USB devices are listed. The usb.cfg file contains a AllUsers section and a section for each user:

```
[AllUsers]
…

[UserA]
…

[UserB]
…

[UserC]
…
```

You can obtain the device identifiers (VID/PID) by launching the script ***ShowExistingUsbDevices.vbs*** (run as administrator). The script generates file ***ExistingUsbDevices.txt***, which contains all installed USB devices in the desired notation. You can paste them directly into your central usb.cfg file. Alternatively you can retrieve the VID/PID identifiers from the USBSecure logfile (USBSecure.log) or from Windows device manager.



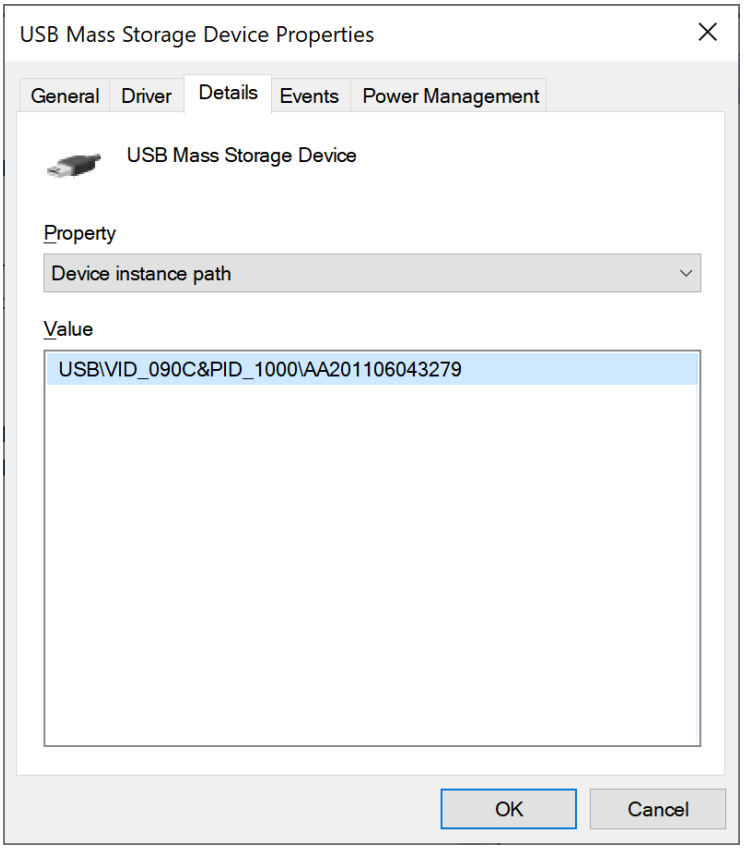
Text behind the hash sign (#) is a comment and will be ignored.

Privileged users with access to any USB device get an asterisk (*):

```
[UserA]
*
```

Please specify USB devices permitted for all users in the [AllUsers] section. You should at least have the line „service = usbhub“ and „service = usbhub3“ in this section. Normally „harmless“ devices like keyboards, mice and scanners are listed here.

You can right-click the USB device in Windows device manager to retrieve the identifier. In this case you would insert **VID_090C&PID_1000\AA201106043279** into usb.cfg to allow exactly this certain USB mass storage device.



It is also allowed to use the term **USB\VID_090C&PID_1000\AA201106043279** with prefix **\USB**, so you can copy the ID directly from device manager.

If you would like to allow any USB mass storage device of the same model or type, you can specify VID_090C&PID_1000.

Allow USB devices per computer

From version 4.4 it is possible to allow USB devices "per computer". This is useful if a specific USB device is connected to a specific computer to which several users log on. Use the following notation for this:

```
[Host:<computername>]
<allowed device>
```

Example:

```
[Host:PC01234]
VID_090C&PID_1000
```

Wildcard question mark

Use the question mark "?" to place a wildcard for a character:

VID_1234&PID_????

This makes it possible to allow all devices from a specific manufacturer: You enter the VID (Vendor ID) and leave the PID (Product ID) variable. For example, to allow all devices from the manufacturer Kyocera, you could use the following expression: VID_0482&PID_????

Case sensitivity

The case is not relevant in all USBSecure configuration files. Entry VID_090C&PID_1000 and entry Vid_090C&Pid_1000 are considered identical.

Services

To enable entire devices classes please use the „service“ command: service=<Service-Name>

Example: `service = usbstor`

Often used values:

service=usbhub	USB root hubs
service=usbhub3	USB3 root hubs
service= iusb3hub	USB3 root hubs
service=hidusb	USB keyboards and mice
service=usbstor	USB mass storage devices (sticks, harddisks…)
service=usbprint	USB printers
service=usbscan	USB scanners

You can determine the service of a USB device in Device manager / Properties of the device / Service.

Devices allowed by default

Since USBSecure Professional version 4.3 certain USB devices are enabled by default – they do not have to be listed in the usb.cfg file. These are devices with one of the following values in the „service“ field: usbhub, usbhub3, iusb3hub or hidusb. This feature avoids total loss of functionality in cases of misconfiguration. If you would like to disable these services anyway, please use the following notation in the AllUsers section:

```
no-defaultservice = <Service-Name>
```

Example: `no-defaultservice = hidusb`

Users

Users can be specified in the following notation:

[domain\user] Domain and user
Example: [lugrain\RalphG]

[user@domain] User and domain
Example: [RalphG@lugrain]

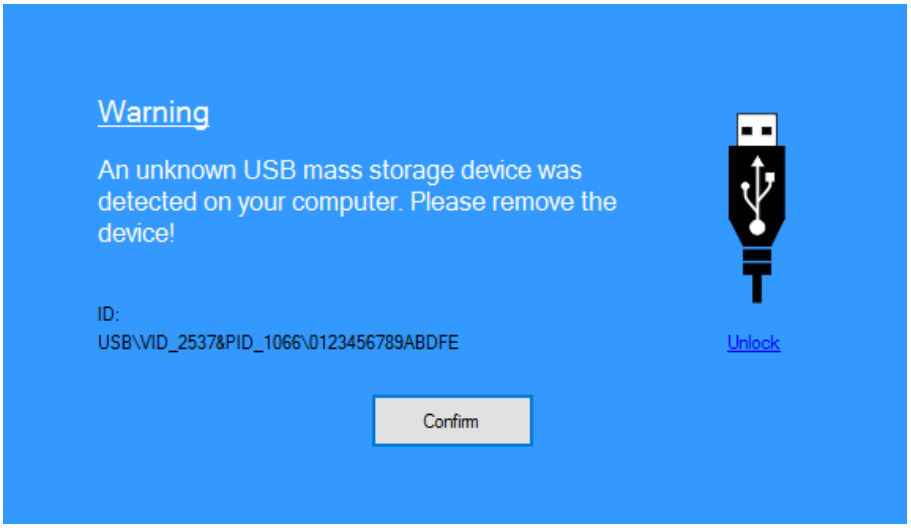
[user] Only the user is listed. The domain is arbitrary. The settings are valid for local and domain users.
Example: [RalphG]

Power Users

PowerUsers are allowed to use all USB mass storage devices (sticks, USB disk drives) temporarily. If a PowerUser inserts a forbidden USB mass storage device, an additional link “Unlock” will be displayed in the blue notification window. This allows the user to enable the mass storage device until the next USBSecure service startup.

Please use the following syntax:

```
[username]
>PowerUser
```



Blacklist services

From USBSecure Professional Version 3.3 you can work with blacklists in usb.cfg. This allows configurations like *"Allow all users any USB device except USB mass storage devices"*. Blacklist services can be applied in the following notation:

```
blacklist-service = <Service-Name>
```

Example: `blacklist-service = usbstor`

Note: Whitelists override blacklists! If a user has the entries **`blacklist-service = usbstor`** and **`service = usbstor`**, USB mass storage devices will be allowed.

Example configurations usb.cfg

You would like to...

...allow only USB keyboards and mice, nothing else:

```
[AllUsers]
service = usbhub    # USB root hubs should always be allowed
service = usbhub3   # USB root hubs should always be allowed
service = hidusb    # any USB keyboard and mouse allowed
```

... allow only USB keyboards and mice, nothing else. Allow a certain USB stick for user miller:

```
[AllUsers]
service = usbhub    # USB root hubs should always be allowed
service = usbhub3   # USB root hubs should always be allowed
service = hidusb    # any USB keyboard and mouse allowed

[miller]
VID_090C&PID_34C7   # VidPid of the stick, see device manager
```

... allow only USB keyboards and mice, nothing else. But allow additionally USB mass storage devices for user smith:

```
[AllUsers]
service = usbhub    # USB root hubs should always be allowed
service = usbhub3   # USB root hubs should always be allowed
service = hidusb    # any USB keyboard and mouse allowed

[smith]
service = usbstor   # USB mass storage devices
service = UASPStor  # USB mass storage devices (newer)
```

... allow only USB keyboards and mice, nothing else. But allow additionally any USB device for user Administrator:

```
[AllUsers]
service = usbhub    # USB root hubs should always be allowed
service = usbhub3   # USB root hubs should always be allowed
service = hidusb    # any USB keyboard and mouse allowed

[administrator]
*
```

...allow any USB device for everyone, but no USB mass storage devices:

```
[AllUsers]
*
blacklist-service = usbstor
blacklist-service = UASPStor
```

... allow any USB device for everyone, but no USB mass storage devices. But allow additionally any USB mass storage device for user Administrator and a certain USB mass storage device for User Smith:

```
[AllUsers]
*
blacklist-service = usbstor
```

```
blacklist-service = UASPStor


[administrator]
service = usbstor   # USB mass storage devices
service = UASPStor  # USB mass storage devices (newer)


[Smith]
VID_090C&PID_34C7   # VidPid of the stick, see device manager
```

bluetooth.cfg

Access to Bluetooth devices is specified in the bluetooth.cfg file. As in the usb.cfg file, there is a AllUsers section and a section per user:

```
[AllUsers]
…

[UserA]
…

[UserB]
…
```

As in the usb.cfg file, it is possible to allow Bluetooth devices per user and per computer. Services can also be allowed. However, there are no blacklist services and no power users, but there are the entries AllowFileTransfer and AllowPanNetwork.

AllowFiletransfer

Use AllowFiletransfer=no to prevent the transfer of files and folders via Bluetooth. A file can be transferred via Bluetooth between two paired devices by right-clicking on the file → Send to → Bluetooth device. AllowFiletransfer=no deactivates the Bluetooth device "Bluetooth Device (RFCOMM Protocol TDI)", which means that file transfer is no longer possible.
Please note: AllowFiletransfer=yes "beats" AllowFiletransfer=no. This makes it possible to globally prevent file transfer (AllowFiletransfer=no in the AllUsers section) and to allow it only for individual users (AllowFiletransfer=yes in the user section).

AllowPanNetwork

Use AllowPanNetwork=no to prevent participation in a PAN (Personal Area Network) network via Bluetooth. A PAN network is a wireless network via the Bluetooth interface. It can be easily set up by clicking on the Bluetooth icon in the system tray → "Join a Personal Area Network". AllowPanNetwork=no deactivates the virtual network card "Bluetooth device (Personal Area Network)", which means that it is no longer possible to create a PAN network.

Please note: AllowPanNetwork=yes "beats" AllowPanNetwork=no. This makes it possible to globally prevent the creation of PAN networks (AllowPanNetwork=no in the AllUsers section) and to allow it only for individual users (AllowPanNetwork=yes in the user section).

All Bluetooth settings only take effect when there is a functional Bluetooth interface. The Bluetooth interface itself is usually a USB device and must therefore be activated in the usb.cfg file.

Bluetooth configuration example 1: Allow all Bluetooth devices, don't allow file transfer

One piece of information in advance: The activation of Bluetooth devices is more complicated than the activation of USB devices. While with USB devices in most cases there is exactly one entry in Device Manager for a device, with Bluetooth it is usually necessary to activate several virtual devices in order to be able to use a real device.
For this reason, this configuration example is very interesting. It prohibits file transfer via Bluetooth - with the least administrative effort.

The aim in this example is to globally allow all Bluetooth devices, but to prevent file transfer via Bluetooth.

First make sure that your Bluetooth interface works. In many cases, a USB device must be activated for this - the actual Bluetooth interface. For example, the device could be named "Intel® Wireless BlueTooth®" or "Broadcom Bluetooth Adapter".

The AllUsers area of the bluetooth.cfg file looks as follows by default:

```
[AllUsers]
*
AllowFiletransfer=yes
AllowPanNetwork=yes
service=UmPass
```

► Step 1: Change the entry that allows file transfer of files and folders:

```
AllowFiletransfer=no
```

This entry prevents the file transfer by *clicking the right mouse button → Send to → Bluetooth device* and by *clicking on the Bluetooth icon in the system tray → Send a File*.

► Step 2: Change the entry that allows the creation of PAN networks:

```
AllowPanNetwork=no
```

This entry prevents setup and participation in a PAN (Personal Area Network) via Bluetooth.

We get the following AllUsers area:

```
[AllUsers]
*
AllowFiletransfer=no
AllowPanNetwork=no
service=UmPass
```

With this configuration, the virtual Bluetooth device "Bluetooth Device (RFCOMM Protocol TDI)" and the virtual network card " Bluetooth device (Personal Area Network)" are deactivated. The goal is achieved to allow all Bluetooth devices, but to prevent file transfer via Bluetooth.

If you want to enable file transfer for individual users or computers, you can use the respective entry with "yes" in the section of the user or computer ("yes" overwrites "no"):

```
[MillerM]
AllowFiletransfer=yes
```

or

```
[host:PC12345]
AllowFiletransfer=yes
```

Bluetooth configuration example 2: Allow a Bluetooth mouse

The aim in this example is to enable the Microsoft Bluetooth Mouse 3600 for all users - all other Bluetooth devices should be forbidden. The example can be transferred to any other Bluetooth mouse.

The AllUsers section of the bluetooth.cfg file looks as follows by default:

```
[AllUsers]
*
AllowFiletransfer=yes
AllowPanNetwork=yes
service=UmPass
```

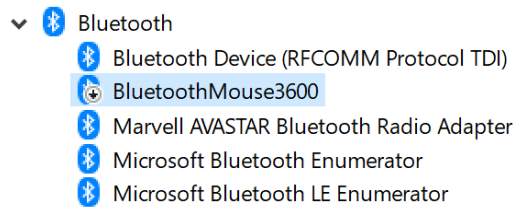
► Step 1: Remove the asterisk (*) that allows all Bluetooth devices for all users. We then receive the following AllUsers area:

```
[AllUsers]
AllowFiletransfer=yes
AllowPanNetwork=yes
service=UmPass
```

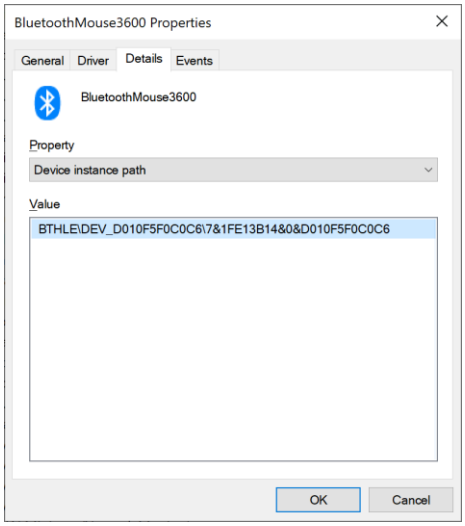
Warning: If you perform this step in your productive environment, Bluetooth devices will no longer work!

► Step 2: Connect the Bluetooth mouse via Settings → Bluetooth and other devices

The Bluetooth mouse is deactivated after connecting.



► Step 3: Enter the device instance path of the "BluetoothMouse3600" device from device manager into the AllUsers section:

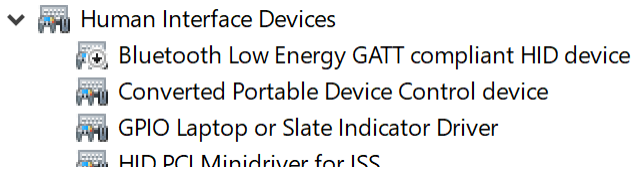


The AllUsers area now looks like this:

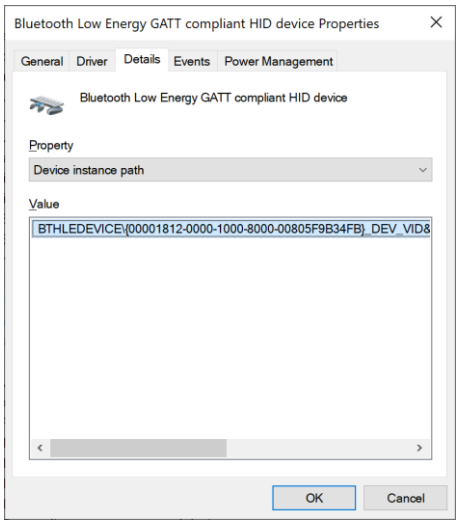
```
[AllUsers]
AllowFiletransfer=yes
AllowPanNetwork=yes
service=UmPass
BTHLE\DEV_D010F5F0C0C6\7&1FE13B14&0&D010F5F0C0C6
```

The "BluetoothMouse3600" device will be enabled after restarting the USBSecure service. However, the mouse still doesn't work.

Under "Human Interface Devices" there is another device that has to be enabled:



► Step 4: Enter the device instance path of the device "Bluetooth Low Energy GATT compliant HID device" from device manager into the AllUsers area.



Only the fixed front part of the device instance path is entered so that all devices of the same type are activated:

```
[AllUsers]
AllowFiletransfer=yes
AllowPanNetwork=yes
service=UmPass
BTHLE\DEV_D007FEE7C0C6\8&18FE6BCF&0&D007FEE7C0C6
BTHLEDEVICE\{00001812-0000-1000-8000-00805F9B34FB}_DEV_VID&02045E_PID&0916_REV&0110
```

BluetoothMouse3600 will be enabled and works after restarting the USBSecure service.

► Problem: The device instance path of the BluetoothMouse3600 device is very variable. Even after removing and reconnecting the mouse, the device instance path changes:

before:

```
BTHLE\DEV_D010F5F0C0C6\7&1FE13B14&0&D010F5F0C0C6
```

after:

```
BTHLE\DEV_D011F6F0C0C6\7&1FE13B14&0&D011F5F1C0C6
```

The problem could be solved with the following entry:

```
BTHLE\DEV_D01?F6F?C0C6\7&1FE13B14&0&D01?F5F?C0C6
```

A question mark (?) stands for one character. However, only the problem for this particular mouse would be solved. Another identical mouse can have a completely different device instance path.

► Solution: Display names can be used in the bluetooth.cfg file. These are the names that can be seen in device manager. In our case, the display name is "BluetoothMouse3600". The complete bluetooth.cfg now looks like this:

```
[AllUsers]
AllowFiletransfer=yes
AllowPanNetwork=yes
service=UmPass
BluetoothMouse3600
BTHLEDEVICE\{00001812-0000-1000-8000-00805F9B34FB}_DEV_VID&02045E_PID&0916_REV&0110
```

Please note that not all display names can be used. To obtain a list of usable display names, please run the VBS file ShowBluetoothDisplaynames.vbs in the USBSecure directory.

Bluetooth configuration example 3: Allow a SmartPhone

The aim in this example is to enable a specific SmartPhone (Android or iPhone) for the AD user MillerM. The connection should be made via Bluetooth - all other Bluetooth devices should be forbidden. The configuration file bluetooth.cfg looks as follows in the standard:

```
[AllUsers]
*
service=UmPass
```

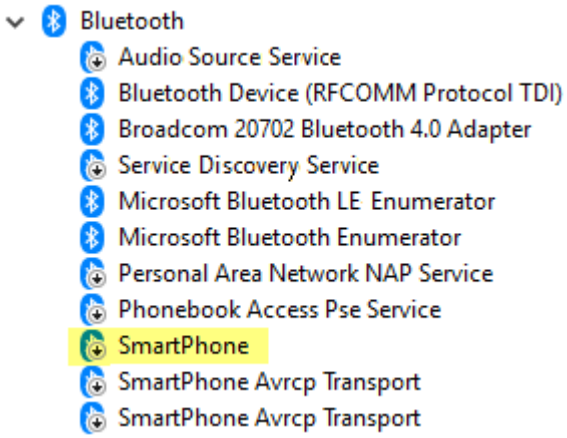
► Step 1: Remove the asterisk (*) that allows all Bluetooth devices for all users. We then receive the following AllUsers area:

```
[AllUsers]
service=UmPass
```

Warning: If you perform this step in your productive environment, Bluetooth devices will no longer work!

► Step 2: Connect the SmartPhone via Settings → Bluetooth and other devices

The SmartPhone will be deactivated after connecting.

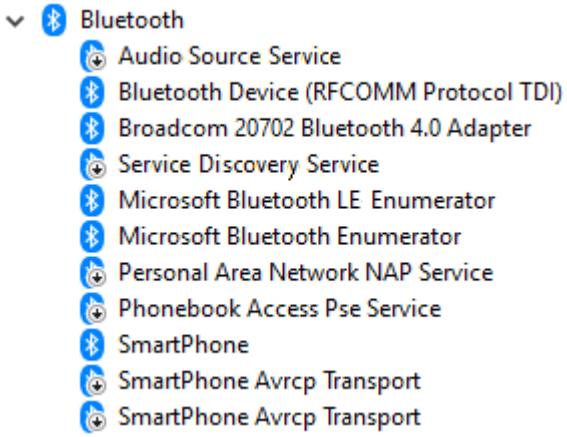


► Step 3: Enter the device instance path of the "SmartPhone" device from device manager in the section for user MillerM:

Configuration file bluetooth.cfg now looks like this:

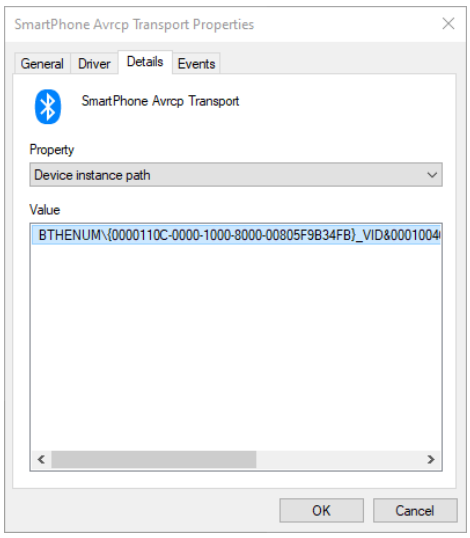
```
[AllUsers]
service=UmPass

[MillerM]
BTHENUM\DEV_103025E38C5C\8&2299B3AE&0&BLUETOOTHDEVICE_103025E38C5C
```

The "SmartPhone" device will be enabled after restarting the USBSecure service. However, there are still some deactivated devices:



Now the device instance paths of all deactivated devices should be entered for MuellerM. But there is also an easier way:

We need the device instance path of one of the deactivated devices, for example that of the "SmartPhone AVRCP Transport" device.



When comparing the device instance paths of all devices that are still deactivated, it is noticeable that they differ only slightly from one another. We replace the deviating numbers with question marks (?):

```
BTHENUM\{????????????????????????????????????}_VID&0001004C_PID&7003\8&2299B3AE&
0&103025E38C5C_C00000000
```

► Step 4: Enter the variable device instance path for all devices that are still deactivated:

```
[AllUsers]
service=UmPass

[MuellerM]
BTHENUM\DEV_103025E38C5C\8&2299B3AE&0&BLUETOOTHDEVICE_103025E38C5C
BTHENUM\{????????????????????????????????????}_VID&0001004C_PID&7003\8&2299B3AE&
0&103025E38C5C_C00000000
```

The SmartPhone is enabled and works after restarting the USBSecure service.

USBSecure.ini

USBSecure.ini contains global settings. Normally you don't have to change these settings.

Server=<name of the USBSecure server>

Specify the server with the devices$ share here (see chapter „Server installation"). This value is automatically inserted during the client installation (MSI package).

LogLevel=<normal|full>

Determines the detail level of logfile USBSecure.log. In production environments this setting should be „normal". Use „full" only for problem analysis.

ViolationReboot=<yes|no>

Defines the behaviour in case of devices that couldn't be deactivated by the operating system without reboot – because they are currently accessed. ViolationReboot=yes means that a reboot is forced.

RebootDelay=60

RebootDelay defines the user's remaining time (in seconds) to save his unsaved documents in case of a ViolationReboot.

RebootMessage=Unregistered USB-, CD/DVD- or Floppy drive detected...

The message that appears in case of a ViolationReboot.

ViolationEject=<yes|no>

Determines if a removable media device (USB stick, CD, …), that couldn't be deactivated by the operating system (because it is currently accessed), should be ejected. An ejected devices must be plugged in again to function for a permitted user.

ScsiSupport=<yes|no>

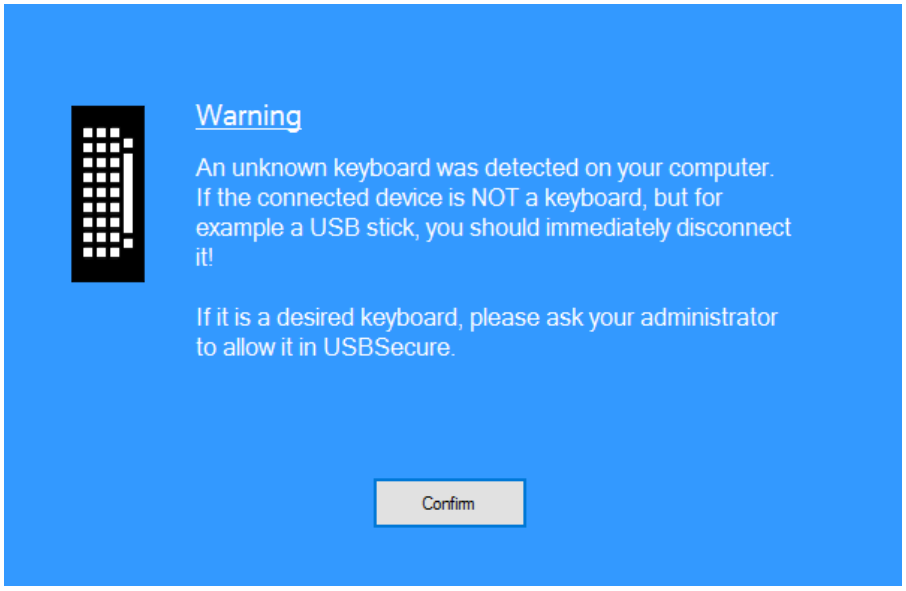
Defines if SCSI CD drives should be supported.

UsbLoggedOffDeactivation=<yes|no>

USB devices not listed in the AllUsers section will be deactivated by default, if no user is logged on. This also happens at startup and shutdown. Under some conditions this behavior is not desired, because a certain USB device is required for the logon process, e.g. a wireless device. With „UsbLoggedOffDeactivation=no" USB devices are not disabled when the user logs off, but only when a new user logs on without permissions.

KeyboardInstall=<block|warn|allow> (default: warn)

USB devices with manipulated firmware can be used to compromise computers by emulating a keyboard (BadUSB). USBSecure detects the number of installed keyboards during its first run (if KeyboardInstall=block or KeyboardInstall=warn is configured) and writes the value into the file KeyboardCount.cfg. If a new keyboard is plugged in, the computer will be locked and the user gets a notification. In mode KeyboardInstall=warn there is just one warning. If the device is a real keyboard it can be used after confirmation by the user. In mode KeyboardInstall=block the computer will be locked permanently until the device is removed.



You can configure the number of allowed keyboards in the local file KeyboardCount.cfg (in the USBSecure folder). Increase the value or delete the file and restart the USBSecure service. Important: In USBSecure.ini KeyboardInstall=warn or KeyboardInstall=block must be configured.

LocalDevicesCopy=30

Determines if the text files containing all installed devices of all users should be stored centrally for inventory. Read/Write share devicesRW$ with folder ExistingUsbDevices must exist (value in minutes, 0=never).

IniOverwrite=60

Controls the central management of file USBSecure.ini. Defines whether file USBSecure.ini should be overwritten by the centrally stored USBSecure.ini (from devices$). Value in minutes, 0=never. Has no effect if the centrally stored USBSecure.ini does not exist.
It is possible to implement different languages for the USBSecure user dialogs (unknown mass storage device and new keyboard) with this setting. Depending on the operating system language, different USBSecure.ini files are copied from the central devices$ share. In the USBSecure.ini file you can define the dialog phrases (see below). Please use the following file names for the language specific USBSecure.ini files: USBSecureLanguage<InstallLanguage>.ini. The <InstallLanguage> value matches the registry value **InstallLanguage** under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\Language. In case of an English operating system the name would be USBSecureLanguage0409.ini.

Example: Your company network consists of english, german and french clients and you would like to implement the USBSecure user dialogs in the specific language. Set parameter IniOverwrite to 5 – this can also be done with MSI parameters during installation. Create the following files in the cetrnal share \\<YourServer>\devices$:
USBSecureLanguage0407.ini, USBSecureLanguage0409.ini und USBSecureLanguage040c.ini. Use the USBSecure.ini from your local installation (C:\Program Files (x86)\USBSecure) as your template. Create german dialog phrases (e.g. TextUsbWarning1, see below) in USBSecureLanguage0407.ini, english dialog phrases in USBSecureLanguage0409.ini and french dialog phrases in USBSecureLanguage040c.ini. The appropriate .ini files will be copied 5 minutes after startup of the USBSecure service (IniOverwrite=5) to the local USBSecure folder.

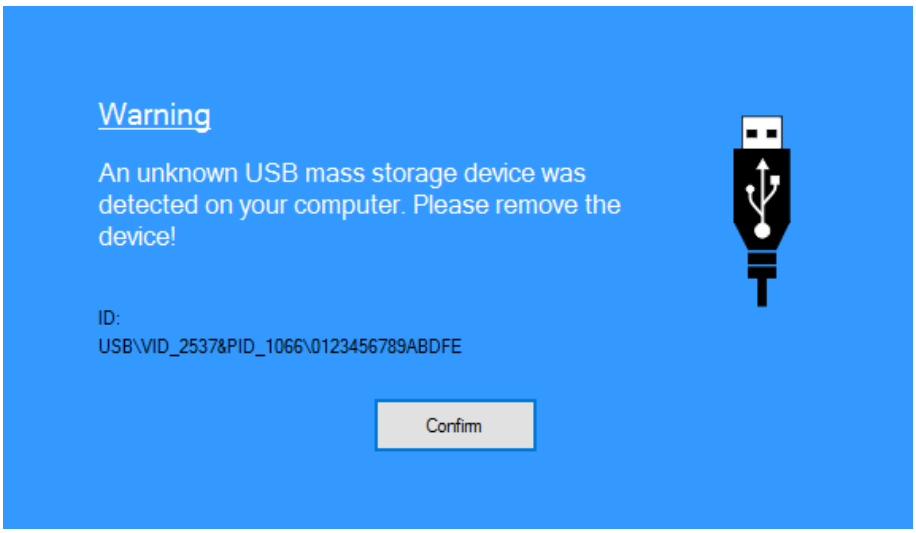
If no language specific USBSecure.ini file exists, USBSecure.ini will be used. If the devices$ share doesn't contain any USBSecure.ini file, nothing happens.

Backup=60

With this setting your local client machine becomes a USBSecure backup machine. All .cfg files and the USBSecure.ini from the devices$ share will be backup up to your local machine into folder **backup** every 60 minutes (value in minutes, 0=never). The files get a time stamp in the filename – old files are not overwritten.

NoUsbStorInfo=<yes|no|warn> (default: no)

When inserting a forbidden USB mass storage, a warning message will be displayed and the computer is locked. Specify NoUsbStorInfo=yes here to surpess this behaviour. USBSecure then behaves like version 3.3 – the USB mass storage device will be disabled only.
The warning message doesn't appear when a mass storage device is already plugged at boot time.



You can set this value during unattended installation with MSI variable NOUSBSTORINFO=yes.

ForceUsbstorUnplug=<yes|no> (default: no)

In a worst case an USB mass storage device can't be disabled by the operating system. If you set ForceUsbstorUnplug=yes, the computer will be locked continuous until the device is unplugged – with a delay of 20 seconds.

ResolveVendors=<yes|no> (default: yes)

With ResolveVendors=no the vendor will NOT be resolved in logfile entries.

SmartphoneInfo=<yes|no> (default: no)

If SmartphoneInfo=yes is configured, the blue warning message will also be displayed when forbidden smartphones, cameras and similar devices are connected to the computer. By default the warning message only appears by connecting USB storage devices (Service USBSTOR). In detail: If SmartphoneInfo ist set to yes, the blue warning message appears if an USB device is connected that has the value WUDFrd or WUDFWpdMtp in the Service field (see Device Manager oder DeviceTool).

UsbStorNotify=<yes|no> (default: no)

When you set UsbStorNotify=yes, a text file will be created in the share devicesRW$\Notify when a forbidden USB mass storage device (stick, hard disk) is plugged in. This information can be used to send notification mails with the SmtpSend.exe utility (bin folder).

UsbCfgSizeCheck=<yes|no> (default: yes)

Since version 4.3, newly copied usb.cfg files are checked for plausibility. If a new usb.cfg is less than 15 bytes large, the new usb.cfg will not be used. It will also not be used if file size is less than 50% of the previous usb.cfg file. In these cases the usb.cfg file from folder "cache" will be used. This bebaviour reduces the risk of misconfiguration. If UsbCfgSizeCheck is set to „no", no plausibility checks will be performed.

SmtpServer=<mailserver name>
MailFrom=<sender>
MailTo1=<recipient1>

MailTo2=<recipient2>
MailTo3=<recipient3>

Specify the sender, recipients and the name of your mailserver here. When a forbidden USB mass storage device (stick, hard disk) is plugged in, these recipients will be notified. Please specify at least values for SmtpServer, MailFrom and MailTo1. These settings are independent of the UsbStorNotify value.

Define your own text to be displayed when a forbidden USB mass storage device is plugged in. „\n“ for a new line:

TextUsbWarning1=Warning
TextUsbWarning2=An unknown USB mass storage device was detected on your computer. Please remove the device!
TextUsbUnlockLink=Unlock
TextUsbConfirmButton=Confirm
TextUsbUnlock1=You are allowed to enable this USB mass storage device temporarily.\n\nNote: This operation will be logged.\n\nWould you like to enable this mass storage device now?
TextUsbUnlock2=Device was enabled. It may be necessary to unplug and reconnect it.
TextUsbMsg=Your computer was locked because of an unknown USB mass storage.\n\nPlease remove the USB device an log on again.

You are able to display a link with the following variables – e.g. a link to your intranet to explain your USB policy:

TextUsbLink=More information
UsbLink=http://intranet.mycompany.com/usb

Define your own text to be displayed when an additional keyboard is connected (if KeyboardInstall=warn or block). „\n“ for a new line:

TextKeyboardWarning1=Warning
TextKeyboardWarning2=An unknown keyboard was detected on your computer. If the connected device is NOT a keyboard, but for example a USB stick, you should immediately disconnect it!\n\nIf it is a desired keyboard, please ask your administrator to allow it in USBSecure.
TextKeyboardUnlockLink=Enable additional keyboard
TextKeyboardConfirmButton=Confirm
TextKeyboardUnlock=You are allowed to enable an additional keyboard.\n\nWould you like to enable the keyboard now?
TextKeyboardMsg=Your computer has been locked because of an unknown keyboard. Please disconnect the most previously connected device!\n\nInformation: USB devices, emulating a keyboard, can be a risk for your computer and your network.

You can display a link with the following settings, e.g. to show additional information about bad USB devices:

TextKeyboardLink=More information
KeyboardLink=http://intranet.mycompany.com/usb

InstallLanguage=0409

USBSecure detects the operating system language automatically – on the basis of the registry value InstallLanguage under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\Language. This value in USBSecure.ini overwrites this setting.

AdminsCantStop=<yes|no>

Determines the right for administrators to stop the USBSecure service. This setting can also be set during Installation (GUI or MSI value ADMINSCANTSTOP=1 or 0). Please note that a change of this value will take 5 minutes (after service startup) to take effect.
Generally an administrator is always able to stop a service. This setting makes it a little bit more difficult.

ApplyConfigAfterServiceStartup=<time in minutes>

When you define ApplyConfigAfterServiceStartup=5, the complete device configuration (usb.cfg, cd.cfg, …) will be copied and reapplied 5 minutes after USBSecure service startup. Use this feature in environments where the network is not ready at service startup, for example in NAC environments (Network Access Control). Possible values are: 2 - 999 (0 = off).

ApplyConfigDailyAt=<time>

Use entry ApplyConfigDailyAt=00:30 to copy and apply the complete device configuration (usb.cfg, cd.cfg, …) at 00:30 AM. Use this feature in environments with computers running 7x24. If a computer is not connected to the network or switched off, this task will not be repeated.

BluetoothSupport=<yes|no>

Use entry BluetoothSupport=no to disable Bluetooth support.

Mail notification

A mail notification can be sent to the USBSecure administrators when a forbidden USB mass storage device (stick, hard disk) is plugged in. While USBSecure has no real server component, mail notification has to be performed directly from the client – or with a central scheduled task.

Mail notification directly from client

→ Required entries in USBSecure.ini: SmtpServer, MailFrom, MailTo1

When a forbidden USB mass storage device is connected, the client sends a notification to the recipients specified in USBSecure.ini (MailTo1, MailTo2 and MailTo3). Communication is established on port 25. Because no authentication takes place, your clients must be enabled for internal relaying on your mailserver. In some environments this method could fail because relaying is not allowed or virus scan software or firewalls could prevent the clients from sending E-Mails.

Centrally managed mail notification

→ Required entries in USBSecure.ini: UsbStorNotify=yes

When a forbidden USB mass storage device is connected, the client creates a text file in folder devicesRW$\Notify. These files can be used to send mail notifications to the USBSecure administrators with a scheduled task.
Please create a scheduled task named “USBSecure Mail” in Computer Management of a Windows Server, running once a minute and performing the following command:

Action: Programm starten
Program/Script: C:\USBSecure\SmtpSend.exe
Add arguments: <mailserver name> <sender> <recipient> "Unknown USB mass storage device (%COMPUTER%)" -folder:"\\<USBSecure servername>\devicesRW$\Notify"

Therefor copy file SmtpSend.exe in a newly created folder C:\USBSecure on the server.

As soon as a forbidden USB mass storage device is plugged in on a client machine a text file will be

created in folder **\\<USBSecure servername>\devicesRW$\Notify**. This text file will be converted into an E-Mail and then moved to folder “done”.

Logfile USBSecure.log

In USBSecure.log all activities are logged (user logons, start of the USBSecure service, enabling/disabling of devices, license messages etc.). The loglevel should be set to „normal“ in production environments. For diagnostic purposes it can be set to „full“. With full logging more detailed information is logged (see chapter USBSecure.ini).

User <LoggedOff>

User <LoggedOff> is logged in the USBSecure.log file any time no user is logged on. User <LoggedOff> can only access the devices listed in the AllUsers section. At the time of logoff (and before logon) all devices not listed in the AllUsers section are disabled. If this behaviour is undesired (e.g. because a USB WLAN adapter is required at logon) you can set „UsbLoggedOffDeactivation=no“ in USBSecure.ini.

Fast User Switching

Fast User Switching was first implemented for domain users in Windows Vista. Fast User Switching allows mutiple users to logon to a Windows machine the same time.
If USBSecure detects that mutiple users have logged on, it turns to „FastUserSwitching“ mode. In „FastUserSwitching“ mode the user-defined allowed devices will be deactivated. Only devices listed in the AllUsers section remain activated.