Lire le SMART d'un HDD / SSD en USB

Problème

Dans certaines situations, le pilote Linux `uas` désactive les transferts SAT (ATA passthrough), ce qui empêche `smartmontools` (et d'autres outils comme `hdparm`) de communiquer avec un disque SATA connecté en USB.

Le symptôme typique est une erreur de ce type :

```
Read Device Identity failed: scsi error unsupported field in scsi command
A mandatory SMART command failed: exiting.
```

Ou encore :

```
Unknown USB bridge [0xXXXX:0xXXXX (0xXXXX)]
Please specify device type with the -d option.
```

Pourquoi ?

Certains chipsets bridges USB-SATA ne gèrent pas correctement les commandes SAT en mode UAS. Le noyau Linux active alors automatiquement le flag `NO_ATA_1X` pour ces chipsets, ce qui désactive le passthrough SAT et bloque smartmontools.

Les appareils concernés incluent notamment les boîtiers Seagate, les chipsets Initio INIC-3069 et VIA VL711.

---

1 - Identifier les IDs USB du périphérique

```
lsusb
```

Repérez la ligne correspondant à votre disque, par exemple :

```
Bus 002 Device 003: ID XXXX:YYYY Nom du fabricant
```

- `XXXX` = vendor ID
- `YYYY` = product ID

Notez ces deux valeurs, elles seront utilisées dans toutes les commandes suivantes.

---

2 - Vérifier le pilote utilisé

```
lsusb -t
```

Si votre périphérique utilise le pilote `uas`, c'est probablement la cause du problème.

3- Forcer le type de device avec smartctl

Avant tout, vérifiez le nom du disque via la commande suivante :

```
sudo lsblk
```

Identifiez le nom du disque que vous voulez diagnostiquer.

> Remplacez `/dev/sdX` par le chemin réel de votre disque (`/dev/sda`, `/dev/sdb`, etc.).

> ⚠ Attention, une mauvaise identification de disque et vous diagnostiquerez le mauvais disque !

Rentrez la commande suivante :

```
sudo smartctl -d sat -a /dev/sdX
```

Si ça échoue, tentez avec une longueur de commande différente :

```
sudo smartctl -d sat,12 -a /dev/sdX
sudo smartctl -d sat,16 -a /dev/sdX
```

---

4 - Basculer temporairement vers usb-storage

Si les commandes ci-dessus échouent, désactivez temporairement UAS pour votre périphérique :

```
sudo modprobe -r uas
sudo modprobe usb-storage quirks=XXXX:YYYY:u
```

Débranchez puis rebranchez le disque, puis relancez smartctl :

```
sudo smartctl -d sat -a /dev/sdX
```

5- Solutions permanentes

Option A — Désactiver UAS (recommandé)

Crée un fichier de configuration modprobe :

```
sudo nano /etc/modprobe.d/usb-storage-quirks.conf
```

Ajoutez :

```
options usb-storage quirks=XXXX:YYYY:u
```

Le flag `:u` active `IGNORE_UAS` — le noyau utilisera `usb-storage` à la place de `uas`.

Option B — Forcer le SAT passthrough en gardant UAS

> ⚠ À vos risques et périls — peut causer des instabilités sur certains chipsets.

Même fichier, mais avec le flag `:t` qui force le passthrough SAT malgré UAS :

```
options usb-storage quirks=XXXX:YYYY:t
```

Pour plusieurs périphériques

Séparez les entrées par des virgules :

```
options usb-storage quirks=XXXX:YYYY:u,AAAA:BBBB:u
```

Mise à jour de l'initramfs

Après avoir créé ou modifié le fichier de configuration :

```
sudo update-initramfs -u
```

Option via GRUB

Modifiez `/etc/default/grub` et ajoutez le paramètre à `GRUB_CMDLINE_LINUX_DEFAULT` :

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash usb-storage.quirks=XXXX:YYYY:u"
```

Puis mettez à jour GRUB :

```
sudo update-grub
```

Référence des flags

Flag	Valeur hex	Lettre	Effet
`IGNORE_UAS`	`0x00800000`	`:u`	Désactive UAS, bascule sur usb-storage
`NO_ATA_1X`	`0x02000000`	`:t`	Force le SAT passthrough (risqué)

Vérification

Après redémarrage ou reconnexion du disque :

```
sudo smartctl -a /dev/sdX
```

Un résultat valide commencera par une section `=== START OF INFORMATION SECTION ===` avec les informations de votre disque.

Sources

- smartmontools Wiki — SAT with UAS under Linux
- Kernel Parameters Documentation

Révision #7
Créé 2026-05-16 10:01:11 CEST par vplur
Mis à jour 2026-05-21 10:37:03 CEST par vplur