



Surprise removal of 1394 storage devices

Steve Powers

Program Manager - 1394

Microsoft

18 January 2001



Problem - Surprise Removal of 1394 Storage Devices

- ***Easily unplugged***

Users commonly remove hardware without first warning the operating system. This action is known as “surprise remove”.

- ***Was it safe?***

While surprise removal appears intuitive and safe to end users, it can result in serious problems, even on hot-pluggable buses.

- ***Data corruption/loss***

Due to write-caching, surprise removal of 1394 storage devices can result in data loss or corruption.

- ***Pressure to fix quickly***

OEMs and Microsoft Product Support want this problem addressed. Microsoft needs to make a decision for our next product release.



What can be done?

- ***Few ways to prevent***

There are limited steps OSVs can take to minimize the potential for data loss/corruption in storage surprise remove scenarios.

- ***Disabling write caching***

Data is immediately written to the storage device whenever a change is made to a file on that device, versus edits being cached and written later.

- ***This affects performance***

Disabling write caching can have a significant performance impact on high performance storage devices such as 1394 hard drives.



Tradeoffs to be made

- ***Safety versus speed***

Disabling write caching may be acceptable for low performance, low capacity external storage in the following cases.

- **Bulk transfers**

User scenario which may not require performance gains provided by write caching (i.e. a user that only does a bulk reads or writes to or from a 1394 storage device but does not do ongoing file edits)

- **High level data integrity**

User scenario where safe surprise removal is valued over optimal performance (i.e. data backup to 1394 storage device)



Tradeoffs to be made...

- ***Speed versus safety***

In other instances, the benefits of increased performance outweigh the probability of data loss.

- **Internal devices**

Category of devices mounted within the PC, which may not be removed without dismantling the system.

- **Educated users**

User scenarios of 1394 drives which demand optimal performance (ex. a professional graphic artists editing graphic files on a 1394 HDD).



Vendors know best

- ***HDDs have many different uses***

Each device and application have their own performance/data integrity requirements.

- ***Vendors should recommend default caching policy which best suits their system or device***

Storage device vendors know how their device is intended to be used. PC OEMs will want a say in how to setup their systems.

- ***OS must be informed***

The OS should know and obey the selected caching policy. There is no way to do this today.



Proposal

- ***Define removal mechanism***

Define a mechanism for peripheral vendors to specify a device's "Removal Policy", the two options being *ExpectOrderlyRemoval* and *ExpectSurpriseRemoval*. This mechanism would be placed into the device's configuration ROM.

This policy would be placed upon the root directory of the module as future device classes other than storage may need to make "caching-like" tradeoffs based on the device's expected usage model.

- ***Two options per IEEE 1212 specification***

- Key Value
- Extended key sequence



Preferred option – KeyValue

- KeyValue 0x30 is reserved for a new "UsageModel" tag. Per IEEE 1212 spec, values 0x30-0x37 are available for use by the bus committee (e.g. 1394)

00 110000 rrrrrrrr rrrrrrrr rrrrrrrr rrrrrrrR - "Usage Model"

Bits 31-1	Reserved for future 1394 committee definitions(should return 0)
Bit 0	Removal Policy (suggests caching policy), 0 for ExpectSurpriseRemoval, 1 for ExpectOrderlyRemoval



Alternate option – extended key sequence

- Extended key sequence used to specify new UsageModel tag (same semantics as above). Per IEEE 1212 spec, all extended sequences are defined using KeyId's 1C, 1D, and 1E.

00 011100 ssssssss ssssssss ssssssss, where s is specifier ID or RID (possibly 0 for SIG, or Microsoft's RID)

00 011101 kkkkkkkk kkkkkkkk kkkkkkkk, where k is key ID specifies which of the specifier's tags is being discussed). A new k is defined for "UsageModel"

00 011110 rrrrrrrr rrrrrrrr rrrrrrrr rrrrrrrrR, where as above r is reserved and R means Removal Policy, 0 for ExpectSurpriseRemoval, 1 for ExpectOrderlyRemoval



Multi-node devices – policy default

- If an OS using this information determines that several nodes are all part of one module, the entire module's removal policy should be *ExpectOrderlyRemoval* if there exists any node within the module containing that value. If no node specifies any value, the OS will fall back to a default removal policy, which for 1394 will be *ExpectSurpriseRemoval*.



Summary

- ***Industry-wide problem***

Surprise removal of hardware is an industry wide problem, even for hot-pluggable buses like 1394 and USB. IHVs and OSVs need a standard policy that allows vendors to specify a default caching policy based on how a device is likely to be used.

- ***Industry-wide compatibility***

Microsoft believes that this solution will work for all OSVs. Devices, which adhere to this proposed specification, may easily be moved between platforms.

- ***Timeframe***

Microsoft would like to implement this quickly. We would like industry approval to resolve this issue in an expeditious manner.