

Project Proposal for A New Standard

1 Source of the Proposed Project

1.1 Title

Protected Area Run Time Interface Extension Services (PARTIES-2)

1.2 Date Submitted

Feb 18, 2003

1.3 Proposer(s)

T13

2 Process Description for the Proposed Project

2.1 Project Type (Development or Revision)

D

2.2 Type of Document

Standard

2.3 Definitions of Concepts and Special Terms

The terms are industry standard.

2.4 Expected Relationship with Approved Reference Models, Frameworks, Architectures, etc.

The standard addresses closed systems and has no relationship to INCITS Reference Models.

2.5 Recommended INCITS Development Technical Committee

T13

2.6 Anticipated Frequency and Duration of Meetings

T13 presently meets up to six times per year and authorizes ad hoc meetings as warranted for the needs of the projects. It is anticipated that these meetings are adequate to address this technical report among the other agenda items for these meetings.

2.7 Target Date for Initial Public Review (Milestone 4)

April 2003

2.8 Estimated Useful Life of Standard or Technical Report

5 years or more

3 Business Case for Developing the Proposed Standard

3.1 Description

This proposal builds on PARTIES to address the following:

- Mandate command usage (28-bit/48-bit) on basis of IDENTIFY DRIVE command result (word 83[10]: “Command sets supported.” and 86[10]: “Command set/feature enabled.”).
- Clarify 5.2.23, device name. Is it device id string copied from IDENTIFY DRIVE command result or is this string defined by BEER creator? **Moreover**, restriction of only using 20h-7Eh (ASCII) makes string English centric and can’t be used for other languages.

Proposal is to remove this restriction. Proposal is to make it “*This is a null terminated string that is suitable for display to the user. If the string length is 40 bytes the null is not present. This string can only be made up of printable characters for current system language, e.g. for English, only ASCII 20h-7Eh characters can be used*”.

- Clarify section 5.2.20 “Number of entries in the BEER Directory of Services” and section 5, Initialization Requirements “*The Directory of Services immediately follows the BEER data and may contain up to six entries. The six entries at 64 bytes each plus the 128 BEER bytes compose the 512 bytes in the last sector on the drive*”.

Section 5 should be modified to “*The Directory of Services immediately follows the BEER data and may contain up to n entries (BEER bytes 80, 81). The first Beer Directory of Service entry starts immediately after the BEER header and if length of the BEER exceeds the available space on the sector it is continued at the start of the preceding sector. For example, for this revision, first six entries at 64 bytes each plus the 128 BEER bytes compose the 512 bytes in the last sector on the drive. Rest “n-6” DOS entries occupy preceding sectors incrementally*”.

- Clarify the meaning of “elimination” of HPA (sec 6.20, last paragraph) in ATA/PI spec 6/rev 3a. One connotation is that HPA has to be *removed* via SET MAX [EXT] (non-volatile) (native max [EXT]) to eliminate it. Another connotation is that simply doing a SET MAX [EXT] (volatile) (native max [EXT]) would suffice. Clarify the meaning to be discussed.

This proposal builds on PARTIES to add the following:

- At present there is no provision for a PSA provider to communicate custom PSA attributes to an upper layer application. For example, consider a PSA that is processed /invoked from an OS based OEM application/driver. This PSA might contain important data for that app/driver. If this application needs some PSA specific attributes, such as its security level or platform specific flags etc. then, at present, there is no way to do it.

Proposal:

Section 5.3, BEER Directory of Services description, Table 2, Byte 1 [3, 0] should be reserved for custom attributes for this PSA, such as security level or OEM flags for PSA.

1	Byte	Bit	Description
		7..4	Reserved
		3..0	OEM flags

- Section 5.3, BEER Directory of Services description, Table 2.

Bytes 26-27 (sec 5.3.6) give Service Area ID, which is provider/vendor’s PCI ID.

There is no possible way to uniquely identify the implementer ID (an implementer is somebody, an OEM or ISV or IT dept or end user, who actually places that PSA in HPA). PSA's might come from different providers. Therefore, we need an ID that uniquely identifies the implementer of that PSA.

This would allow a *provider* application to identify all PSA's provided by that *provider* (using Service Area ID) as well as an *implementer* application to identify all PSA's placed in by that *implementer* only (using implementer ID).

It can be used in conjunction with the PSA OEM flags in byte 1 [3, 0]. An OEM app can identify PSA(s) put in by that OEM and then use OEM flags to determine individual characteristics for that PSA. For example, a PSA (purchased from a provider) can be customized by the IT department according to its requirements and put in the systems along with custom flags to identify security clearance for that system. Now an app installed by IT can use custom methods to find its PSA's (using implementer ID) and then use OEM flags to get other custom PSA information.

String would solely be used to give the NAME of PSA and is used only for presentation purposes.

Proposal:

To use bytes 60-61 for implementer ID.

Table 2 – BEER Directory of Services

60-61	Word	OEM/Implementer ID
-------	------	--------------------

Addition:

5.3.8 OEM/Implementer ID

The OEM ID is used to enable OEM, IT department as well as end-user to place PSA's on the drive later on during lifetime of the system. The ID shall be the same code allocated to a vendor for the purposes of PCI identification. If the vendor does not have a PCI identification number then this field is cleared to 0.

- In current spec when a PSA is opened, the only level of protection is DOS Byte 0[4], "Service area is read only". This protection is software based and can easily be overridden by malicious code. This inherently makes the 'secure' area 'insecure'. We need a provision to prevent or at least hinder the malicious code from destroying the opened area within the HPA space.

One possible way is to always make the opened area within the HPA read only. However, this is not a viable solution since an application running in a PSA may need to write to this area.

The following list details various cases of HPA availability and respective security state:

1. When a "SET MAX ADDRESS" command has not been used to establish an HPA, the whole disk constitutes the User Area. Therefore there is no protected space on the hard disk.
2. Assume a "SET MAX ADDRESS" command is used to create an HPA, and the drive is left unlocked. In this case, malicious code can open the HPA space by doing a SET MAX ADDRESS to NATIVE MAX, thereby compromising the data within the HPA. In this state, the HPA state is 'unprotected'. A SET MAX LOCK is required to change the HPA state to 'protected'.
3. Assume a "SET MAX ADDRESS" command is used to create an HPA, and the drive is locked using the "SET MAX LOCK" command. The HPA cannot be opened unless the drive

is subsequently unlocked using a predetermined password. Therefore, in this state, the HPA state is protected.

4. To execute a PSA from a 'protected' HPA, a "SET MAX UNLOCK" followed by a volatile "SET MAX ADDRESS" command is used to temporarily open the HPA area up to this PSA's boundary. At this time, the opened HPA area can be compromised. Even though the whole HPA is intended to be in a 'protected' state, the opened HPA area is in an 'unprotected' state.

Proposal:

To implement a new flag to be passed during UNLOCK command. UNLOCK command is used to open a '*secured*' area and therefore is best candidate to inform hard disk firmware what to do with this open secured area.

If this flag is SET, then HDD firmware makes the area from non-volatile set max (start of HPA) to volatile SET MAX (end of open HPA) as RO (read only). This is backwards compatible because current applications keep this flag cleared to 0 (reserved bit) and therefore by default the area would be opened as RW, just as it is done today. To set this bit or not would be decided on the basis of bit DOS Byte4[0] "Service area is Read Only".

This flag doesn't cover all security risks, but does cover most cases when the PSA being opened only executes an application (reads) and doesn't need to write anything (e.g. an anti virus application might write its log into C: drive instead of PSA).

This flag takes effect right after the UNLOCK command. If an HPA area is already open then its attribute is changed to reflect this UNLOCK command's flag. If HPA size is shifted using volatile SET MAX then the new opened HPA keeps the same attribute. In other words, any opened HPA area attribute is same as set via last UNLOCK command.

Best case scenario would arise from a configuration where all read only PSA's have been clubbed together near start of HPA, therefore opening up any of these PSA's would open HPA in RO mode and therefore HPA is open but still secured.

Register	7	6	5	4	3	2	1	0
Device	obs	RO FLAG	obs	DEV	Start LBA (27:24)			

At present this flag is cleared to 0 because, as per current spec, this is NA.

This flag is in effect till next UNLOCK command, where it can be changed.

Change:

5.3.1.2 Service Area is Read Only

When this bit is set to one no data shall be written to the HPA when this PSA is opened. This makes both this PSA as well as 'other' opened HPA space secure from writing. This field is intended as a user flag and shall be enforced by the OS, BIOS and the drive F/W. It is possible for the user to set this bit to 0, write new data to the service area, and set the bit back to 1.

3.2 Existing Practice and the Need for a Standard or Technical Report

The Standard is needed to provide guidance on methods that will facilitate use of the reserved area in a manner that does not cause interoperability issues.

Some large companies have already announced diagnostic boot capability for their products and that they would work with T13 to arrive at an industry compatible method for this function. Numerous companies

are shipping notebook computers that utilize the reserved area for saving the system context and memory when the notebook goes into hibernation. These various uses need to be harmonized. The Standard should allow existing implementations to be accommodated with a reasonable migration plan, potential applications to be implemented through the next year while documenting harmonious methods which will be stable for more than three years.

3.3 Implementation Impacts of the Proposed Standard or Technical Report

3.3.1 Development Costs

Implementation costs are born on a voluntary basis by industry. Members of T13 have informed us that their companies consider the detailed costs to be confidential information. But the cost is considered to be fairly modest as the standard describes existing BIOS practices. Logistical costs for T13 are negligible since the technical report represents about 5% of the T13 meeting agenda. Although the members consider the cost details to be confidential, they also consider the ultimate costs to be reduced by the benefits of the technical report.

3.3.2 Impact on Existing or Potential Markets

3.3.3 Costs and Methods for Conformity Assessment

No formal conformity assessment is undertaken. However each of the personal computer systems manufacturers have extensive qualification testing which on a voluntary basis assures the methods are exhaustively tested in the industry. The incremental cost is modest.

3.3.4 Return on Investment

The estimated ROI for development of this technical report and the conformity assessment costs associated with it greatly exceeds 1000 to 1.

3.4 Legal Considerations

3.4.1 Patent Assertions

T13 will make regular calls for patents in the meetings addressing the technical report. There are none at this time

3.4.2 Dissemination of the Standard

Drafts of this document will be disseminated electronically. Dissemination of the final standard will be restricted, as the document becomes the property of INCITS, ANSI, or ISO/IEC.

4 Related Standards Activities

4.1 Existing Standards

ANSI NCITS 346-2001 (PARTIES)
T13/1321D (ATA/ATAPI-5)
NCITS TR-21 (EDD)

4.2 Related Standards Activity

T13/d1410r3b (ATA/ATAPI-6)

T13/d1532v1r1a, d1532v2r1a (ATA/ATAPI-7)

4.3 Recommendations for Coordinating Liaison

None

4.4 Recommendations for Close Liaison

T10