ISO/IEC 11002

Edition 1.0 2008-07

INTERNATIONAL STANDARD

Information technology – Multipath management (API)

ISO/IEC 11002:2008(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub

The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

ISO/IEC 11002

Edition 1.0 2008-07

INTERNATIONAL STANDARD

Information technology – Multipath management (API)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE W

ICS 35.200

ISBN 2-8318-9908-7

CONTENTS

INFORMATION TECHNOLOGY – MULTIPATH MANAGEMENT (API)

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 11002 was prepared by the Information Technology Industry Council and was adopted under the fast track procedure by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

The Multipath Management application programming interface (API) provides management interfaces to standard capabilities defined in ISO/IEC 14776-453 (SPC-3) and common vendor-specific extensions to the standard capabilities. The intended audience is vendors that deliver drivers that provide these capabilities. This standard relates to SCSI multipathing features and excludes multipathing between interconnect devices (such as Fibre Channel switches) and transport specific multipathing (such as iSCSI multiple connections per session).

INFORMATION TECHNOLOGY – MULTIPATH MANAGEMENT API

1 Scope

This International Standard provides management interfaces to standard capabilities defined in ISO/IEC 14776-453 (SPC-3) and common vendor-specific extensions to the standard capabilities. The intended audience is vendors that deliver drivers that provide these capabilities. This standard relates to SCSI multipathing features and excludes multipathing between interconnect devices (such as Fibre Channel switches) and transport specific multipathing (such as iSCSI multiple connections per session).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

The provisions of the referenced specifications other than ISO/IEC, IEC, ISO and ITU documents, as identified in this clause, are valid within the context of this International Standard. The reference to such a specification within this International Standard does not give it any further status within ISO or IEC. In particular, it does not give the referenced specification the status of an International Standard.

ISO/IEC 9899:1999, *Programming languages – C*

ISO/IEC 14165-133, *Information technology – Fibre channel – Part 133: Switch fabric-3 (FC-SW-3)*

ISO/IEC 14165-251, *Information technology – Fibre channel – Part 251: Framing and signalling (FC-FS)*

ISO/IEC 14776-115, *Information technology – Small computer system interface (SCSI) – Part 115: Parallel interface-5 (SPI-5)*

ISO/IEC 14776-150, *Information technology – Small computer system interface (SCSI) – Part 150: Serial attached SCSI (SAS)*

ISO/IEC 14776-413, *Information technology – Small computer system interface (SCSI) – Part 413: Architecture model-3 (SAM-3)*

ISO/IEC 14776-453, *Information technology – Small computer system interface (SCSI) – Part 453: Primary commands-3 (SPC-3)*

RFC 3720, *Internet Small Computer Systems Interface (iSCSI)*

NOTE Copies of IETF standards such as RFC 3720 may be obtained through the Internet Engineering Task Force (IETF) at http://www.ietf.org.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document the following terms and definitions apply.

3.1.1
auto-failback
capability of some multipath drivers to resume use of a path when the path transitions from unavailable to available

3.1.2
auto-probing
capability of some multipath drivers to validate operational paths that are not currently being used

3.1.3
available paths
set of paths for a logical unit that may be considered for routing I/O requests

NOTE For symmetric access devices, all paths are considered *available*. For asymmetric access devices, all paths in active target port groups are considered *available*.

3.1.4
device file
operating system files (for instance UNIX, Linux etc.,) that facilitate communication with the system's hardware and peripherals

3.1.5
device identification VPD page
VPD page that provides the means to retrieve identification information about the SCSI device, logical unit, and SCSI port

3.1.6
hexadecimal-encoded binary data
ASCII character string used to denote the hexadecimal encoding of a binary string of octets

NOTE It may only contain the ASCII characters 0-9, A-F, and a-f. Two hexadecimal characters represent each byte of binary data.

3.1.7
host
compute node connected to the SAN

3.1.8
initiator
SCSI device that initiates requests, also known as a client

NOTE In this document, initiator refers to an initiator port.

3.1.9
logical unit
addressable entity within a SCSI target

NOTE For example, RAID arrays expose each virtual disk volume as a logical unit. When the term "logical unit: is used in this standard and is not qualified as a "multipath logical unit" or "path logical unit", it refers to a logical unit in a target device.

3.1.10
multipath logical unit
an object type of this API representing a "virtual "logical unit that coalesces multiple path logical units for the same underlying device logical unit

3.1.11
object ID
unique identifier assigned to any object within the MP API

NOTE Objects sometimes represent physical entities, e.g. initiator ports. At other times, objects represent logical entities, e.g. target port groups.

3.1.12
path
association between an initiator port, target port and logical unit, see 3.1.13

3.1.13
path logical unit
an object type of this API providing access to a single logical unit through a single initiator port and single device port

NOTE Within this API, each path (see 3.1.12) is modelled as a path logical unit. The result of multipath drivers is a single OS device file representing a multipath logical unit aggregating multiple path logical units.

3.1.14
persistent
quality of something being non-volatile

NOTE This usually means that the associated data is recorded on some non-volatile medium such as flash RAM or magnetic disk and that the data survives beyond system reboots. Implicitly, the data is readable from the non-volatile medium.

NOTE Examples of persistent storage:

- under Windows, the registry would be a common place to find persistently stored values (assuming that the values are not stored as volatile);
- under any OS a file on magnetic hard disk would be persistent.

3.1.15
plugin
software, written for an OS, HBA or device vendor that provides support for one or more multipath drivers

NOTE The plugin's job is to provide a bridge between the library's interface and the vendor's multipath driver. A plugin is typically a loadable module, for instance, a DLL in Windows and a shared object in UNIX. A plugin is accessed by an application through the Multipath Management API library.

3.1.16
product (or device product)
a particular model of target device, identified by the vendor, product and revision IDs returned in the standard SCSI INQUIRY command response

3.1.17
target
SCSI device containing logical units and SCSI target ports that receives commands from a SCSI initiator

3.1.18
target port group
set of target ports that are in the same target port access state at all times

3.1.19
unicode
system of uniquely identifying (numbering) characters such that nearly any character in any language is identified.

3.1.20
VPD
vendor specific information about a device returned in response to a SCSI INQUIRY command with the EVPD bit set (see SPC-3)

3.2 Abbreviations

API	application programming interface
DLL	dynamic link library
HBA	host bus adapter
LUN	logical unit number
OID	object identifier
OS	operating system
UML	Unified Modeling Language
UTF	Unicode Transformation Format
VPD	vital product data

4 Document conventions

The API is specified as a set of types and structures (see Clause 6) followed by a set of function definitions (see Clause 7). This clause discusses the formats used in these clauses along with conventions used in defining the API.

Constants are defined as a list of #defines followed by a typedef for a C integer type. C language enums do not have a specific size; using #defines rather than enums helps assure client code is interoperable across platforms and compilers – especially if used in C++ applications.

API description format

Each API's description is divided into seven subclauses.

1 **Synopsis**

This subclause gives a brief description of what action the API performs.

2 **Prototype**

This subclause gives a prototype of the function in a format that is a combination of a C function prototype and an Interface Definition Language (IDL) prototype. The prototypes show the following:

- the name of the API;
- the return type of the API;
- each of the parameters of the API, the type of each parameter, and whether that parameter is an input parameter, output parameter, or both an input and an output parameter.

3 **Parameters**

This subclause lists each parameter along with an explanation of what the parameter represents.

4 **Typical return values**

This subclause lists the Typical Return Values of the API with an explanation of why a particular return value would be returned. It is important to note that this list is not a comprehensive list of all of the possible return values. There are certain errors, e.g. MP_STATUS_INSUFFICIENT_MEMORY, which might be returned by any API.

5 **Remarks**

This subclause contains comments about the API that may be useful to the reader. In particular, this subclause contains extra information about the information returned by the API.

6 **Support**

This subclause states that if an API is mandatory to be supported, optional to be supported, or mandatory to be supported under certain conditions.

- If an API is mandatory to be supported a client can rely on the API functioning under all circumstances.
- If the API is optional to be supported then a client cannot rely on the API functioning.
- If the API is mandatory to be supported under certain conditions then a client can rely on the API functioning if the specified conditions are met. Otherwise a client should assume that the API is not supported.

7 **See also**

This subclause lists other related APIs or related code examples that the reader might find useful.

5 Background technical information

5.1 Overview

Open system platforms give applications access to physical devices by presenting a special set of file names that represent the devices. Although end users typically don't use these special device files, knowledgeable applications (file systems, databases, backup software) operate on these device files and provide familiar user interfaces to storage. The device files have a hierarchical organization, either by using files and directories or by naming conventions.

This hierarchy of device files (sometimes called a device tree) provides an effective interface for simpler, desktop device configurations. Inside open systems kernels, the hierarchy is exploited to allow different drivers to operate on different parts of the device tree. When the OS discovers connected devices and builds the device tree, multiple paths to the same device may show up as separate device files in the device tree. Separate storage applications using device files that represent paths to the same device will overwrite each other's data.

As storage products (typically disk arrays) strove for better reliability and performance, they added multipath support. For OSes that lacked multipath support, the device and logical volume manager vendors provided multipath drivers. Device standards lacked standard interfaces for identifying multipath devices; so multipath drivers are often limited to specific device products. Recently standards have been defined and OS vendors have started integrating multipath support in their bundled drivers.

These drivers create special device files that represent multipath devices. Storage applications like file systems can use these multipath device files the same way they would use a single-path device file, but benefit from improved reliability and performance. In addition, the multipath drivers provide some management capabilities, for example, failover or load balancing, that only apply to multipath devices.

This standard focuses on devices accesses through SCSI commands. SCSI commands are sent to a target device by an initiator. The target may consist of multiple logical units. For example, a RAID array exposes virtual disk as separate logical units. A target device supporting multiple paths and attached hosts will nearly always have multiple ports. Each permutation of initiator port, target port and logical unit is commonly referred to as a path. With no multipath support in place, the OS would see each path as separate logical units. The function of multipath drivers is then to create a virtual multipath device that aggregates all these path logical units.

5.2 Target port groups

A logical unit may only be accessible through certain target ports. If the device supports asymmetric access (see 5.3.2), certain ports may be preferred for access (sometimes this is referred to as affinity. ISO/IEC 14776-453 (SPC-3) has introduced target port groups as a way for target devices to represent access characteristics for logical units. A target port group is a collection of ports. All the logical units associated with that target port group share the same access state (active/optimized, active/non-optimized, standby or unavailable).

Target port groups are abstract elements that may or may not equate to an element of the target system (such as a controller).

The concept of target port groups can be applied to all devices, even if they don't actually implement the SCSI standard interfaces. This API does not require an SPC-3-compliant array; it includes target port groups and uses the terminology of ISO/IEC 14776-453 (SPC-3) as a starting point, but is extended to reflect common vendor implementations.

In order to simplify tasks for client software, all plugins/drivers make it appear that the underlying hardware uses target port group interfaces. For example, consider an asymmetric array with two ports where each port is primary (optimized) for half the logical units. The plugin/driver would create four "virtual" target port groups; each logical unit would be part of two target port groups, one with optimized access state for its primary controller and one with non-optimized access state for the secondary controller. See Annex B for more details.

5.3 Relationship between target port groups in SCSI and in this API

5.3.1 General

This subclause describes the relationship between the interfaces defined in ISO/IEC 14776-453 (SPC-3) and this API related to target port groups.
The SCSI **Device Identification VPD page (i.e., page 83h)** and REPORT TARGET PORT GROUPS command allow initiators to discover the target port group configuration.

- **Device Identification VPD page** returns a list of identifiers. These include:
    - **relative target port identifier** – a two-byte value with a target-unique ID for the target port the INQUIRY is sent to. In this API, this is the relativePortID property in MP_TARGET_PORT_PROPERTIES;
    - **target port group identifier** – a two-byte value with a target-unique ID for the target port group. In this API, this is the tpgID property of MP_TARGET_PORT_GROUP_PROPERTIES.
- The REPORT TARGET PORT GROUPS command returns a list of target port groups, with access state, and the list of relative port Ids of target ports that comprise each target port group. The access state corresponds to this API's MP_ACCESS_STATE_TYPE and MP_TARGET_PORT_GROUP accessState property.

The SCSI SET TARGET PORT GROUPS command allows an initiator to set target port access state – which causes failover or failback. This API provides MP_SetTPGAccess as an interface to SET TARGET PORT GROUPS.
For a concrete example, Figure 1 depicts a RAID array with asymmetric access and two controllers. Each controller contains two ports that always share the same access state. The RAID configuration is set up with four logical units. Optimally each pair of logical units is accessed through the ports on different controllers. In case either controller fails, all four logical units can be accessed through the ports in the alternate controller.

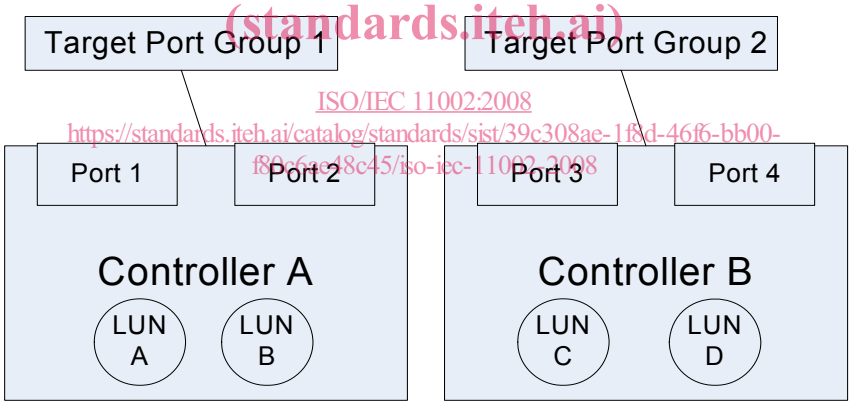

Figure 1 – Asymmetric array example

The table below summarizes the information returned for this array configuration in the SCSI INQUIRY identifiers and REPORT TARGET PORT GROUPS command response.

Logical unit	Access from port 1 or 2	Access from port 3 or 4
	TPG ID / State	TPG ID / State
A	1 / Active optimized	2 / Standby
B	1 / Active optimized	2 / Standby
C	1 / Standby	2 / Active optimized
D	1 / Standby	2 / Active optimized

In case of a failure condition of controller A, all logical units as accessed from port 1 or 2 will either see lack of response or a TPG access state of unavailable. Logical units A and B as seen through ports 3 or 4 will see an access state of active non-optimized.