The value assigned to I/O Class field in Table B.7 of the SRP document needs to be changed as a result of changing the format of this component in the latest InfiniBand^{TM} Identifiers Annex, which is a supplement to InfiniBand^{TM} Architecture Specification Volume 1.

0.1 Comment

Need to change the value of I/O Class field in Table B.7 to comply with the new format specified in the latest InfiniBand^{TM} Identifiers Annex. The relevant text from the latest version of the Annex is included in Section 0.3.

0.2 Description

The format of IOCProfile:Class,SubClass components has been changed in the latest revision of the InfiniBand^{TM} Identifiers Annex to identify the category of I/O driver. The values defined for I/O Class and I/O SubClass components in Table B.7 of the SRP document are 0xFF00 and 0x609E (0x00609E is the OUI for X3)respectively and follows the format (See Figure 1:) of Class, SubClass components from the previous version of InfiniBandTM Identifiers Annex.The most significant byte of the I/O Class field needs to be changed to comply with the new format specified in Figure 2: .

Figure 1: Previous format of Class/SubClass fields

Field:	Class Field		SubClass Field
Bits:	16 8	7 0	16
Content:	0xFF	OrganizationID[a]	

[a] The most significant byte of the OUI is in the least significant byte of the Class field and the least significant byte of the OUI is in the least significant byte of Subclass field. Bits are in canonical order.

0.3 Text from InfiniBand^{TM} Identifiers Annex

0.3.1 I/O Controller Identification

The DevMgt class IOControllerProfile attribute contains components that identify an I/O controller.These components are: VendorID, DeviceID, Device Version, Subsystem VendorID, SubsystemID, IO Class, IO Subclass, Protocol, Protocol Version, ID String. In addition, each I/O controller has at least one associated ServiceEntries attribute.

The I/O controller vendor provides this information so the host can identify the I/O controller and match it with an appropriate I/O driver (refer to the I/O annex). This section describes the acceptable practice for deriving values for these fields.

0.3.2 Vendor Information

The following components are vendor specific: VendorID, DeviceID, Device Version, Subsystem VendorID, SubsystemID, ID String.

The vendor places its IEEE assigned Organization Unique Identifier (OUI) in the VendorId field and may place any value in the DeviceID and Device Version fields. The vendor may also provide an ASCII string of its choice in the ID String field.

The Subsystem VendorID and SubsystemID provide additional information when a subsystem vendor uses components provided by other vendors. In this case the subsystem vendor provides its OUI in the Subsystem VendorID field and may specify any value in the SubsystemID field.

A vendor that produces a generic controller (i.e., one that supports a standard I/O protocol such as SRP), which does not have vendor specific device drivers, may use the value of 0xFFFFFF in the VendorID field. However, such a value prevents the vendor from ever providing vendor specific drivers for the product.

0.3.3 Generic Information

Generic information fields refer to the IO Class, IO Subclass, Protocol, and Protocol Version components of the IoControllerProfile. An I/O Controller (IOC) uses these fields to indicate that it supports a standard I/O protocol[1]. A host uses these fields to match the IOC with an I/O driver that performs that I/O protocol (see I/O Annex for driver matching).

An IOC shall specify IOCProfile: Class, Subclass, and Protocol values in accordance with Section 0.3.1 I/O Controller Identification.

It is the combination of the Class, Subclass, and Protocol fields that identifies a single I/O class protocol.

The most significant byte of the Class field identifies the I/O category and the least significant byte of the Class field and the SubClass field identify the defining organization as illustrated inFigure 2:.

[1] A standard I/O protocol refers to a protocol definition that permits a vendor other than the IOC vendor to provide the I/O driver. Thus, vendor and product specific information is not pertinent in matching the IOC with an I/O driver.

Figure 2: Class/Subclass fields for External Protocols

Field:	Class Field			SubClass Field
Bits:	15-12	11-8	7-0	15-0
Content:	Category Table 1	reserved	OrganizationID[a]	

[a] The most significant byte of the OUI is in the least significant byte of the Class field and the least significant byte of the OUI is in the least significant byte of Subclass field. Bits are in canonical order.

Table 1 specifies the values for Category:

Table 1 I/O Category

Value	I/O Category
0x0	none or other
0x1	Storage class
0x2	Network class
0x4	Video/Multimedia class
0xF	Unknown or multiple classes
others	reserved

0.3.4 IBTA Protocols

An OrganizationID code of 0xFFFFFF indicates an IBA defined class protocol as listed in Table 2.

Table 2 IBTA Defined Protocols

Class[a]	Subclass	Protocol[b]	Definition
0xzzFF	0xFFFF	0xFFFF	The IOC does not support a standard I/O protocol (only a proprietary protocol)
0x40FF	0xFFFF	0x0001	Console protocol as defined in Console Annex

[a] z=any value

[b] All other combinations reserved

0.3.5 Other Protocols

Any organization may define their own protocol identifiers by specifying a Class, SubClass, Protocol tuple using their OrganizationID in the Class field (i.e., the least significant 8 bits) and the Subclass field, as illustrated in Figure 2. The OrganizationID is the IEEE Organization Unique Identifier (OUI) assigned to the organization that is defining the Protocol and Version field values. Thus any organization that has an IEEE assigned OUI may specify Protocol codes. For each Protocol value, the defining organization selects a Category. It also maintains control of defining Version values.