

CEN

CWA 14050-31

WORKSHOP

July 2007

AGREEMENT

ICS 35.240.15; 35.200; 35.240.40

English version

**Extensions for Financial Services (XFS) interface specification -
Release 3.03 - Part 31: XFS MIB Device Specific Definitions -
Identification Card Device Class MIB 1.1**

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

© 2007 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.:CWA 14050-31:2007 D/E/F

Table of Contents

FOREWORD	3
1. INTRODUCTION	5
2. XFS IDC MIB VARIABLES	7
2.1 XFS IDC STATUS TABLE.....	7
2.1.1 <i>xfSIDCStatusTable: States</i>	7
2.2 XFS IDC SUB DEVICE TABLE	9
2.3 XFS IDC ERROR TABLE.....	9
2.4 XFS IDC RESET TABLE.....	10
2.5 XFS IDC RESET DEVICE TABLE.....	10
3. IDC TRAPS	12
3.1 IDC DETAILED DEVICE STATUS CHANGE TRAP.....	12
3.1.1 <i>IDC Detailed Device Status Change Trap Format</i>	12
3.1.2 <i>IDC Detailed Device Status Change Trap: an example</i>	14
3.2 IDC SUB-DEVICE STATUS CHANGE TRAP	15
3.3 IDC RESET DEVICE COMPLETE TRAP	15
3.3.1 <i>IDC Reset Device Complete Trap Format</i>	15
3.3.2 <i>IDC Reset Device Complete: an example</i>	17
4. APPENDIX A - IDC MIB SUB-TREE	19
4.1 IDC MIB IN SMIV2 AND SMIV1 ASN-1 FORMAT	19
5. APPENDIX B - C-HEADER FILES	27
5.1 XFSMIBIDC.H.....	27

Foreword

This CWA is revision 3.03 of the XFS interface specification.

The CEN/ISSS XFS Workshop gathers suppliers as well as banks and other financial service companies. A list of companies participating in this Workshop and in support of this CWA is available from the CEN/ISSS Secretariat.

This CWA was formally approved by the XFS Workshop meeting on 2004-09-24. The specification is continuously reviewed and commented in the CEN/ISSS Workshop on XFS. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this revision 3.03.

This document supersedes CWA 14050-31:2004.

The CWA is published as a multi-part document, consisting of:

Part 1: Application Programming Interface (API) - Service Provider Interface (SPI); Programmer's Reference

Part 2: Service Classes Definition; Programmer's Reference

Part 3: Printer Device Class Interface - Programmer's Reference

Part 4: Identification Card Device Class Interface - Programmer's Reference

Part 5: Cash Dispenser Device Class Interface - Programmer's Reference

Part 6: PIN Keypad Device Class Interface - Programmer's Reference

Part 7: Check Reader/Scanner Device Class Interface - Programmer's Reference

Part 8: Depository Device Class Interface - Programmer's Reference

Part 9: Text Terminal Unit Device Class Interface - Programmer's Reference

Part 10: Sensors and Indicators Unit Device Class Interface - Programmer's Reference

Part 11: Vendor Dependent Mode Device Class Interface - Programmer's Reference

Part 12: Camera Device Class Interface - Programmer's Reference

Part 13: Alarm Device Class Interface - Programmer's Reference

Part 14: Card Embossing Unit Class Interface - Programmer's Reference

Part 15: Cash In Module Device Class Interface- Programmer's Reference

Part 16: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 17: Printer Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 18: Identification Card Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.00 (see CWA 14050-4:2000; superseded) - Programmer's Reference

Part 19: Cash Dispenser Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 20: PIN Keypad Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.00 (see CWA 14050-6:2000; superseded) - Programmer's Reference

Part 21: Depository Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 22: Text Terminal Unit Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 23: Sensors and Indicators Unit Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.01 (this CWA) - Programmer's Reference

Part 24: Camera Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 25: Identification Card Device Class Interface - PC/SC Integration Guidelines

Part 26: Identification Card Device Class Interface - Migration from Version 3.0 (see CWA 14050-4:2000; superseded) to Version 3.02 (this CWA) - Programmer's Reference

Part 27: PIN Keypad Device Class Interface - Migration from Version 3.0 (see CWA 14050-6:2000; superseded) to Version 3.02 (this CWA) - Programmer's Reference

Part 28: Cash In Module Device Class Interface - Migration from Version 3.0 (see CWA 14050-15:2000; superseded) to Version 3.02 (this CWA) - Programmer's Reference

Part 42: PIN Keypad Device Class Interface - Migration from Version 3.02 (see CWA 14050-6:2003; superseded) to Version 3.03 (this CWA) - Programmer's Reference

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is available online from <http://www.cenorm.be/iss/Workshop/XFS>.

The following parts constitute an optional addendum to this CWA. They define the integration between the SNMP standard and the set of status and statistical information exported by the service providers.

Part 29: XFS MIB Architecture and SNMP Extensions – Programmer's Reference

Part 30: XFS MIB Device Specific Definitions - Printer Device Class

Part 31: XFS MIB Device Specific Definitions - Identification Card Device Class

Part 32: XFS MIB Device Specific Definitions - Cash Dispenser Device Class

Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class

Part 34: XFS MIB Device Specific Definitions - Check Reader/Scanner Device Class

Part 35: XFS MIB Device Specific Definitions - Depository Device Class

Part 36: XFS MIB Device Specific Definitions - Text Terminal Unit Device Class

Part 37: XFS MIB Device Specific Definitions - Sensors and Indicators Unit Device Class

Part 38: XFS MIB Device Specific Definitions - Camera Device Class

Part 39: XFS MIB Device Specific Definitions - Alarm Device Class

Part 40: XFS MIB Device Specific Definitions - Card Embossing Unit Class

Part 41: XFS MIB Device Specific Definitions - Cash In Module Device Class

Part 43: XFS MIB Device Specific Definitions – Vendor Dependent Mode Device Class

Part 44: XFS MIB Application Management

The information in this document represents the Workshop's current views on the issues discussed as of the date of publication. It is furnished for informational purposes only and is subject to change without notice. CEN/ISSS makes no warranty, express or implied, with respect to this document.

Revision History:

1.0	20 January 2004	Initial release of XFS MIB specification
1.1	15 April 2007	Update of the MIB to add support for a Detailed Status Trap, a Device Reset capability and the support of SMIV2.

This CEN Workshop Agreement is publicly available as a reference document from the National Members of CEN : AENOR, AFNOR, ASRO, BDS, BSI, CSNI, CYS, DIN, DS, ELOT, EVS, IBN, IPQ, IST, LVS, LST, MSA, MSZT, NEN, NSAI, ON, PKN, SEE, SIS, SIST, SFS, SN, SNV, SUTN and UNI

Comments or suggestions from the users of the CEN Workshop Agreement are welcome and should be addressed to the CEN Management Centre.

1. Introduction

This document provides the device specific MIB definition (Management Information Base) variables for the xfsIDC sub-tree version one, as foreseen by the *XFS MIB Architecture and SNMP Extensions Programmer's reference* document. All the attributes in all the MIBs are Mandatory. In the case where a vendor's device does not support an attribute then a request for this unsupported attribute should return NULL.

The xfsIDC version one sub-tree is identified by:

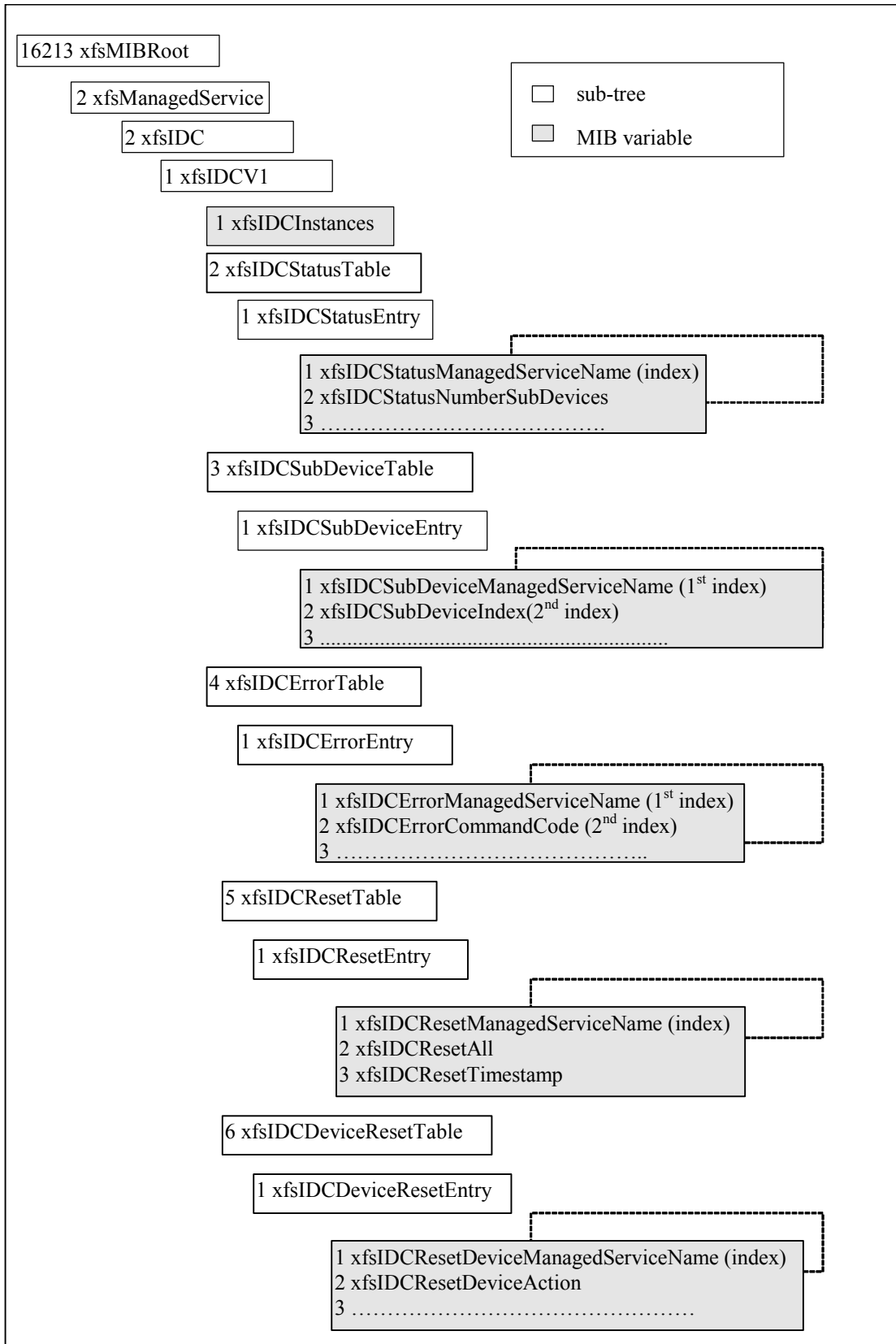
xfsMIBRoot

- xfsManagedService(2)
 - xfsIDC (2)
 - xfsIDCV1 (1)

The xfsIDCV1 sub-tree contains the following variables:

- *xfsIDCInstances(1)* is the number of managed services for the IDC class installed on the XFS subsystem. It is a 32 bit numerical field.
- *xfsIDCStatusTable(2)* identifies the table for the IDC variables.
- *xfsIDCSubDeviceTable(3)* not applicable to the IDC device.
- *xfsIDCErrorTable(4)* identifies the table for the IDC error counters.
- *xfsIDCResetTable(5)* identifies the table for the IDC reset variable.
- *xfsIDCResetDeviceTable(6)* identifies the table for the IDC reset device variables.

The *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document provides an overview of the MIB structure. The following picture shows the structure of the *xfsIDCV1* sub-tree.



Section 2 describes how the Status, Sub-Device, Error and Reset tables apply to the IDC device class.

2. XFS IDC MIB variables

This section describes the MIB variables for the tables of the IDC Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Identification Card Device Class Interface Programmer's Reference*. The following are some general notes pertaining to the MIB variables:

- All command response counters maintained by the service provider are persistent across re-boots.
- One application command may trigger only one command-related counter to be updated.
- One application command may trigger one or multiple status variables to be updated.
- All command response counters are read-writable unless otherwise specified.
- Each managed service has a Reset table that allows all the response counters to be reset.
- Each managed service has a Reset Device table that allows the WFS_CMD_IDC_RESET command to be executed from the management station.

2.1 XFS IDC Status Table

The `xfsIDCStatusTable` groups the variables identifying device status information, statistics and auxiliary variables. It is indexed through a single parameter, `xfsIDCStatusManagedServiceName`. All device status variables are read-only.

Additional variables can be used to contain vendor-dependent variables. These variables do not start immediately after the standard variables in order to allow for expansion of the standard variables, the first additional variable can be added at position 1000.

`xfsIDCStatusManagedServiceName` is the instance identifier of the managed service and uniquely identifies one instance of the IDC class.

As an example, the identifier for the device status value of `xfsIDCStatusMedia(4)` for a device with managed service name equal to "CardReader1" is as follows:

Character	C	a	r	d	R	e	a	d	e	r	l
ASCII Hex	43	61	72	64	52	65	61	64	65	72	31
ASCII Dec	67	97	114	100	82	101	97	100	101	114	49

NOTE: SNMP OID representation of strings consists of a length field specifying the number of characters in the string followed by the ASCII code in decimal for each character in the string. Therefore the OID of the above example is:

`xfsMIBRoot.2.2.1.2.1.4.11.67.97.114.100.82.101.97.100.101.114.49`

2.1.1 xfsIDCStatusTable: States

The first three status variables are common across all device classes, the other variables are device class specific.

`xfsIDCStatusManagedServiceName (1)`

Uniquely identifies the managed service.

`xfsIDCStatusNumberSubDevices (2)`

Defines how many sub-devices the service has. This is always 0 in the IDC device class.

`xfsIDCStatusDevice (3)`

It contains the device state. It is a numeric type field. Allowed values are:

Value	Meaning
<code>xfsDevOnline(1)</code>	The device is present, powered on and online (i.e., operational, not busy processing a request and not in an error state).

xfsDevOffline(2)	The device is offline (e.g., the operator has taken the device offline by turning a switch or pulling out the device).
xfsDevPowerOff(3)	The device is powered off or physically not connected.
xfsDevNoDevice(4)	There is no device intended to be there; e.g. this type of self service machine does not contain such a device or it is internally not configured.
xfsDevHWError(5)	The device is present but inoperable due to a hardware fault that prevents it from being used.
xfsDevUserError(6)	The device is present but a person is preventing proper device operation. The application should suspend the device operation or remove the device from service until the service provider generates a device state change event indicating the condition of the device has changed e.g. the error is removed (WFS_IDC_DEVONLINE) or a permanent error condition has occurred (WFS_IDC_DEVHWERROR).
xfsDevBusy(7)	The device is busy and unable to process an Execute command at this time.

xfsIDCStatusMedia (4)

It contains the media state. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCMediaPresent(2)	Media is present in the device, not in the entering position and not jammed.
xfsIDCMediaNotPresent(3)	Media is not present in the device and not at the entering position.
xfsIDCMediaJammed(4)	Media is jammed in the device; operator intervention is required.
xfsIDCMediaNotSupported(5)	Capability to report media position is not supported by the device (e.g., a typical swipe reader).
xfsIDCMediaUnknown(6)	The media state cannot be determined with the device in its current state (e.g., the value of <i>fwDevice</i> is WFS_IDC_DEVNODEVICE, WFS_IDC_DEVPOWEROFF, WFS_IDC_DEVOFFLINE, or WFS_IDC_DEVHWERROR).
xfsIDCMediaEntering(7)	Media is at the entry/exit slot of a motorized device.
xfsIDCMediaLatched(8)	Media is present & latched in a Latched-DIP card unit.

xfsIDCStatusRetainBin (5)

It contains the state of the ID card unit retain bin. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCRetainBinOK(2)	The retain bin of the ID card unit is not full.
xfsIDCRetainBinNotSupported(3)	The ID card unit does not support retain capability.
xfsIDCRetainBinFull(4)	The retain bin of the ID card unit is full.
xfsIDCRetainBinHigh(5)	The retain bin of the ID card unit is nearly full.

xfsIDCStatusSecurity (6)

It contains the state of the security unit. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCSecurityNotSupported(2)	No security module is available.
xfsIDCSecurityNotReady(3)	The security module is not ready to process cards.
xfsIDCSecurityOpen(4)	The security module is open and ready to process cards.

xfsIDCStatusCardRetained (7)

It contains the number of cards retained. It is a numeric type field. The number of cards retained; applicable only to motor driven ID card units for non-motorized card units this value is 0.

xfsIDCStatusChipPower (8)

It contains the state of the chip controlled by this service. It is a numeric type field. Allowed values as follows

Value	Meaning
xfsIDCChipOnline(1)	The chip is present, powered on and online (i.e. operational, not busy processing a request and not in an error state).
xfsIDCChipPoweredOff(2)	The chip is present but powered off (i.e. not contacted).
xfsIDCChipBusy(3)	The chip is present, powered on and busy (unable to process an Execute command at this time).
xfsIDCChipNoDevice(4)	A card is currently present in the device, but has no chip.
xfsIDCChipHwError(5)	The chip is present, but inoperable due to a hardware error that prevents it from being used (e.g. MUTE, if there is an unresponsive card in the reader).
xfsIDCChipNoCard(6)	There is no card in the device.
xfsIDCChipNotSupported(7)	The capability to report the state of the chip is not supported by the device.
xfsIDCChipUnknown(8)	The state of the chip cannot be determined with the device in its current state.

xfsIDCStatusExtraStatus (100)

It contains the vendor dependent additional device status information as an OCTET STRING. The information is returned as a series of "key=value" strings. Each string is null-terminated, with the final string terminating with two null characters.

2.2 XFS IDC Sub Device Table

The IDC service class does not support any sub-devices, therefore the *xfsIDCStatusNumberSubDevices* will be reported as zero. Sub-device tables are usually used to report sub-device status for Cash Units within a CDM or CIM device class.

2.3 XFS IDC Error Table

The *xfsIDCErrorTable* provides access to all command response counters supported by a device class. The error table contains the set of counters for every combination of executable command and associated response that the service provider supports. The counters report the number of times that a response has been returned from a particular command since the counts were last reset. Selection of the required counter is made by specifying the managed service name, command code and response code through the following parameters:

xfsIDCErrorManagedServiceName
xfsIDCErrorCommandCode
xfsIDCErrorResponseCode

The *xfsIDCErrorTable* is defined as:

- *xfsIDCErrorManagedServiceName(1)* which provides the primary index to the service in question. It is Display String field. The *xfsIDCErrorManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "CardReader1".
- *xfsIDCErrorCommandCode(2)* is an index which identifies the command code that that response code related to, e.g. WFS_CMD_IDC_READ_TRACK (201). It is a 32 bit numerical field.
- *xfsIDCErrorResponseCode(3)* is an index which identifies the response code that the count is required for. It is the absolute value of the error code e.g. WFS_ERR_IDC_MEDIAJAM (-200) is represented by 200. It is a 32 bit numerical field
- *xfsIDCErrorCount(4)* is the count of the number of times that a particular response code has been generated while executing a specific command, since they were last reset. It is a 32 bit numerical field.

All counter variables are read-write. Issue of a Set command on a specific counter with value x will result in the individual counter being set to value x.

As an example, the identifier for the error count value for the WFS_ERR_IDC_MEDIAJAM(-200) error returned from the WFS_CMD_IDC_READ_TRACK(201) command for a device with managed service name equal to "CardReader1" is as follows:

xfsmIBRoot.2.2.1.4.1.4. 11.67.97.114.100.82.101.97.100.101.114.49.201.200.

2.4 XFS IDC Reset Table

The *xfsmIBResetTable* contains the *xfsmIBResetAll* and *xfsmIBResetTimestamp* variables and is indexed by the single variable, *xfsmIBResetManagedServiceName*. When the *xfsmIBResetAll* variable is set to 0 (zero), all the counters in the error table for the managed service are reset to 0 (zero), all other values are ignored.

The *xfsmIBResetTable* is defined as:

- *xfsmIBResetManagedServiceName(1)* which provides the index to the service in question. It is Display String field. The *xfsmIBResetManagedServiceName* parameter corresponds to the value of *xfsmIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "CardReader1".
- *xfsmIBResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsmIBResetAll* variable with value 0 (zero) will result in all counters for the managed service being reset to value 0 (zero). Any other value will be ignored. A query of the *xfsmIBResetAll* variable will return 0 (zero).
- *xfsmIBResetTimestamp(3)* is a read-only variable which represents the UTC date and time when the counters in the error table was reset, it is a Display String field. The data is formatted in the following way: "DD/MM/YYYY HH:MM:SS +ZZZ" where DD/MM/YYYY HH:MM:SS is the local date and time. ZZZ is the bias, which is the difference, in minutes, between Co-ordinated Universal Time (UTC) and local time.

As an example, all the error counts can be reset for a device with managed service name equal to "CardReader1" by setting the value zero in the *xfsmIBResetAll* variable represented by:

xfsmIBRoot.2.2.1.5.1.2. 11.67.97.114.100.82.101.97.100.101.114.49

2.5 XFS IDC Reset Device Table

The *xfsmIBResetDeviceTable(6)* is indexed by the single variable, *xfsmIBResetDeviceManagedServiceName*. This table contains variables which monitor and control the execution of the reset request.

The *xfsmIBResetDeviceAction* variable is used to initiate a reset. Setting this variable will cause the following to happen

1. The SNMP agent will determine if a Device Reset is allowed by checking the *RemoteDeviceResetAllowed* configuration flag (see XFS Common Management Configuration section, within the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document). If it is not allowed then the flow continues with step 5, otherwise the flow continues with step 2.
2. Exclusive access to the device will be obtained
3. A WFS_CMD_IDC_RESET command will be issued
4. Exclusive access to the device will be relinquished when the WFS_CMD_IDC_RESET command completes. Note: Exclusive access must be relinquished as soon as possible and implemented in such a way that deadlocks are avoided.
5. A *xfsmIBResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request

The *xfsmIBResetDeviceMediaControl* variable is used to report how any media found within the device is handled.

The *xfSIDCResetDeviceTable(6)* is defined as:

- *xfSIDCResetDeviceManagedServiceName(1)* which provides the index to the service in question. It is a Display String field. The *xfSIDCResetDeviceManagedServiceName* parameter corresponds to the value of *xfMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. “CardReader1”.
- *xfSIDCResetDeviceAction(2)* is a read-write variable. Issue of a Set command on the *xfSIDCResetDeviceAction* variable with value *executeReset(1)* will result in the device being reset as described above.
- *xfSIDCResetDeviceMediaControl(3)* is a read-only variable. This variable reports how any media found within the device is handled. The value of the *xfSIDCResetDeviceMediaControl* variable is configured through the *ResetDeviceMediaControl* configuration setting (see Managed Service Configuration section, within the *XFS MIB Architecture and SNMP Extensions Programmer’s Reference* document). If this value is not configured then the variable defaults to the *mediaDefault* value that indicates that the Service Provider is responsible for media control.
- *xfSIDCResetDeviceStatus(4)* is a read only variable This variable can be used to check if a reset operation is still in progress. It is set when the reset is initiated and cleared when the reset command completes.

As an example, the device with managed service name equal to “CardReader1” is reset by setting the *xfSIDCResetDeviceAction* variable represented by:

```
xfMIBRoot.2.2.1.6.1.2.11.67.97.114.100.82.101.97.100.101.114.49
```

3. IDC Traps

The following sections define XFS Traps that are specific to the IDC device class.

3.1 IDC Detailed Device Status Change Trap

Status changes within managed services are reported as system events to the XFS Agent. The following section explicitly defines the format of the IDC Detailed Device Status Change trap. However, the format is split into two sections; the fields that are common to all device specific traps and the fields that are specific to each device class. The common fields are defined in the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document. The fields that are specific to the IDC reflect the IDC Status Table as defined in section 2.1.

The detailed device status change event is only generated when the top level status changes within a managed service, i.e. the trap is generated when the fwDevice value in the WFS_INF_IDC_STATUS response has changed. In addition, this trap is only generated on version 1.1 of the MIB and higher and is sent in addition to the summary device status change trap.

The SNMP Specific trap value 102 defines the trap as an IDC Detailed Device Status Change trap.

3.1.1 IDC Detailed Device Status Change Trap Format

The following defines the variable bindings included in the IDC Detailed Device Status Change Trap.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName (1)

This variable binding contains the system generating the alarm, it is a Display String field. It corresponds to lpszWorkstationName in the device status change event data from the service provider.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName (2)

This variable binding represents the managed service name generating the alarm, it is a Display String field. The agent derives this field from the device status change event.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass (3)

This variable binding represents the XFS service class identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<ManagedServiceName>\class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CDM=3, etc. See the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document for a complete list of these values.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName (4)

This variable binding represents the XFS service class name generating the alarm, it is a Display String field. It corresponds to the three character representation of the XFS device class name, and it is useful for human interpretation of a trap. The class name is identified from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<ManagedServiceName>\class.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType (5)

This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS_INF_IDC_CAPABILITIES.fwType field.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid (6)

This variable binding represents the OID of the sub-tree within xfsManagedService defining the management information for this class of managed service. This variable, along with the managed service name as an index, prevents the need for additional querying to find the service specific MIB branch. The IDC MIB class is represented by .1.3.6.1.4.1.16213.2.2

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName (7)

This variable binding represents the physical device name or names associated with the managed service generating the alarm, it is a Display String field. It corresponds to the physical device name or names identified by the managed service. The managed service name is used to identify the physical device name or names, from registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\PhysicalDeviceName. Multiple physical device names are comma separated.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor (8)

This variable binding represents the XFS device vendor name of the device generating the alarm, it is a Display String field. It corresponds to the vendor name for the service provider. The service provider is identified from the managed service name and the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\ServiceProvider.

The service provider name is then used to identify the vendor, from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS*<ServiceProviderName>*\vendor_name.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion (9)

This variable binding represents the XFS MIB version of the device generating the alarm, it is a Display String field. It corresponds to the XFS MIB version for the managed service. The managed service name is used to identify the XFS MIB version, from registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\MibVersion.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent (10)

In case of XFS this variable binding represents the XFS event generating the alarm, it is a 32-bit integer (INT32). It corresponds to u.dwEventID in the event data from the service provider. See the Application Programming Interface (API) - Service Provider Interface (SPI); Programmer's Reference for a complete description of the event structure.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate (11)

This variable represents the UTC and bias for local translation of the date and time when the event was generated. It is a Display String field. The data is formatted in the following way: "DD/MM/YYYY HH:MM:SS +ZZZ" where DD/MM/YYYY HH:MM:SS is the local date and time. ZZZ is the bias, which is the difference, in minutes, between Co-ordinated Universal Time (UTC) and local time.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion (12)

This variable represents the vendor-defined version of the service provider generating the alarm, it is a Display String field. The service provider is identified from the managed service name and the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\ServiceProvider.

The service provider name is then used to identify the version, from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS*<ServiceProviderName>*\version.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.

xfsIDCStatusDevice.xfsIDCStatusManagedServiceName(13)

This variable binding represents the current state of the physical device managed by the service. It is a 32 bit integer (INT32).

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusNumberSubDevices.xfsIDCStatusManagedServiceName** (14)

Defines how many sub-devices the service has. This is the number of retract bins the device supports.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMedia**.xfsIDCStatusManagedServiceName (15)

It contains the media state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusRetainBin.xfsIDCStatusManagedServiceName** (16)

It contains the state of the ID card unit retain bin. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusSecurity.xfsIDCStatusManagedServiceName** (17)

It contains the state of the security unit. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusCardRetained.xfsIDCStatusManagedServiceName** (18)

It contains the number of cards retained. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusChipP
owner.xfsIDCStatusManagedServiceName (19)**

It contains the state of the chip controlled by this service. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusExtra
Status.xfsIDCStatusManagedServiceName (20)**

It contains the vendor dependent additional device status information as an OCTET STRING. The information is returned as a series of "key=value" strings. Each string is null-terminated, with the final string terminating with two null characters.

3.1.2 IDC Detailed Device Status Change Trap: an example

As an example, the following variable binding list represents a detailed device status change trap (6, 102) that is generated for an IDC with a managed service name of "CardReader1". It reports that the device is OFFLINE because the media is jammed.

xfsmIBRoot.3.1.3.1	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName)
	"SST System 1"
xfsmIBRoot.3.1.3.2	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	"CardReader1"
xfsmIBRoot.3.1.3.3	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	2 (WFS_SERVICE_CLASS_IDC)
xfsmIBRoot.3.1.3.4	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	"IDC"
xfsmIBRoot.3.1.3.5	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	3 (WFS_IDC_TYPEDIP)
xfsmIBRoot.3.1.3.6	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	"1.3.6.1.4.1.16213.2.2"
xfsmIBRoot.3.1.3.7	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	"ABC Corp Card Reader"
xfsmIBRoot.3.1.3.8	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	"Best Devices Incorporated"
xfsmIBRoot.3.1.3.9	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	"1.10"
xfsmIBRoot.3.1.3.10	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent)
	4 (WFS_SYSE_DEVICE_STATUS)
xfsmIBRoot.3.1.3.11	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	"20/03/2003 15:40:53 -300"
xfsmIBRoot.3.1.3.12	(xfsmIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	"1.23"
xfsmIBRoot.2.2.1.2.1. 3.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusDevice.xfsIDCStatusManagedServiceName)

	2 (WFS_STAT_DEVOFFLINE)
xfsMIBRoot.2.2.1.2.1.2.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusNumberSubDevices.xfsIDCStatusManagedServiceName)
	0 (No sub device)
xfsMIBRoot.2.2.1.2.1.4.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusMedia.xfsIDCStatusManagedServiceName)
	4 (xfsIDCMediaJammed)
xfsMIBRoot.2.2.1.2.1.5.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusRetainBin.xfsIDCStatusManagedServiceName)
	2 (xfsIDCRetainBinOK)
xfsMIBRoot.2.2.1.2.1.6.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusSecurity.xfsIDCStatusManagedServiceName)
	2 (xfsIDCSecurityNotSupported)
xfsMIBRoot.2.2.1.2.1.7.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusCardRetained.xfsIDCStatusManagedServiceName)
	14 (14 cards retained)
xfsMIBRoot.2.2.1.2.1.8.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusChipPower.xfsIDCStatusManagedServiceName)
	1 (xfsIDCChipOnline)
xfsMIBRoot.2.2.1.2.1.100.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.xfsIDCStatusExtraStatus.xfsIDCStatusManagedServiceName)
	"0"0' (No extra data)

3.2 IDC Sub-Device Status Change Trap

The IDC does not currently support any sub-devices so the IDC Sub-Device Status Change Trap is not currently defined. The SNMP Specific trap value 202 is reserved in case a sub-device is ever added to the IDC device class.

3.3 IDC Reset Device Complete Trap

On the IDC device class this trap reports the completion of the reset device request and includes the status of the device at that point. If the reset has changed the status of the device then the Device Status Change and a Detail Device Status traps will also be generated.

The SNMP Specific trap value 302 defines the trap as a IDC Reset Device Complete trap.

3.3.1 IDC Reset Device Complete Trap Format

The following defines the variable bindings included in the IDC Reset Device Complete Trap. In the following section, the numbers in parenthesis at the end of each binding just indicate the sequence of the variable bindings within the trap, they do not represent an OID value.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult (1)

This variable binding contains a value indicating if the reset was executed, and if not provides a reason. It does not report the status of the device (i.e. the result of the reset), the current status of the device is reported within the **xfsxfsIDCStatusDevice** binding (var bind 12 below).

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName (2)

This variable binding represents the managed service name generating the alarm, it is a Display String field. The agent derives this field from the device status change event.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass (3)

This variable binding represents the XFS service class identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CDM=3, etc. See the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document for a complete list of these values.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName (4)

This variable binding represents the XFS service class name generating the alarm, it is a Display String field. It corresponds to the three character representation of the XFS device class name, and it is useful for human interpretation of a trap. The class name is identified from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\class.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType (5)

This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS_INF_IDC_CAPABILITIES.fwType field.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid (6)

This variable binding represents the OID of the sub-tree within xfsManagedService defining the management information for this class of managed service. This variable, along with the managed service name as an index, prevents the need for additional querying to find the service specific MIB branch. The IDC MIB class is represented by .1.3.6.1.4.1.16213.2.2

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName (7)

This variable binding represents the physical device name or names associated with the managed service generating the alarm, it is a Display String field. It corresponds to the physical device name or names identified by the managed service. The managed service name is used to identify the physical device name or names, from registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\PhysicalDeviceName. Multiple physical device names are comma separated.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor (8)

This variable binding represents the XFS device vendor name of the device generating the alarm, it is a Display String field. It corresponds to the vendor name for the service provider. The service provider is identified from the managed service name and the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\ServiceProvider.

The service provider name is then used to identify the vendor, from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS*<ServiceProviderName>*\vendor_name.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion (9)

This variable binding represents the XFS MIB version of the device generating the alarm, it is a Display String field. It corresponds to the XFS MIB version for the managed service. The managed service name is used to identify the XFS MIB version, from registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\MibVersion.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate (10)

This variable represents the UTC and bias for local translation of the date and time when the event was generated. It is a Display String field. The data is formatted in the following way: "DD/MM/YYYY

HH:MM:SS +ZZZ" where DD/MM/YYYY HH:MM:SS is the local date and time. ZZZ is the bias, which is the difference, in minutes, between Co-ordinated Universal Time (UTC) and local time.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion (11)

This variable represents the vendor-defined version of the service provider generating the alarm, it is a Display String field. The service provider is identified from the managed service name and the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS*<ManagedServiceName>*\ServiceProvider.

The service provider name is then used to identify the version, from the registry value

HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS*<ServiceProviderName>*\version.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.

xfsxfsIDCStatusDevice.xfsIDCStatusManagedServiceName(12)

This variable binding represents the current state of the physical device managed by the service. It is a 32 bit integer (INT32).

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusNumberSubDevices.xfsIDCStatusManagedServiceName (13)**

Defines how many sub-devices the service has. This is the number of retract bins the device supports.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMedia.xfsIDCStatusManagedServiceName (15)**

It contains the media state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusRetainBin.xfsIDCStatusManagedServiceName (16)**

It contains the state of the ID card unit retain bin. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusSecurity.xfsIDCStatusManagedServiceName (17)**

It contains the state of the security unit. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusCardRetained.xfsIDCStatusManagedServiceName (18)**

It contains the number of cards retained. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusChipPower.xfsIDCStatusManagedServiceName (19)**

It contains the state of the chip controlled by this service. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusExtraStatus.xfsIDCStatusManagedServiceName (20)**

It contains the vendor dependent additional device status information as an OCTET STRING. The information is returned as a series of "key=value" strings. Each string is null-terminated, with the final string terminating with two null characters.

3.3.2 IDC Reset Device Complete: an example

As an example, the following variable binding list represents a Reset Device Complete trap (6, 302) generated as a result of a request to reset the device from the remote management station. The device in question has a managed service name "CardReader1".

xfsMIBRoot.3.1.3.1.3	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult)
	0 (resetExecuted)
xfsMIBRoot.3.1.3.2	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	"CardReader1"
xfsMIBRoot.3.1.3.3	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClasses)
	2 (WFS_SERVICE_CLASS_IDC)
xfsMIBRoot.3.1.3.4	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	"IDC"
xfsMIBRoot.3.1.3.5	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	3 (WFS_IDC_TYPEDIP)
xfsMIBRoot.3.1.3.6	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)

	“.1.3.6.1.4.1.16213.2.2”
xfsmIBRoot.3.1.3.7	(xfsmIBRoot.xfsTrap. xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	“ABC Corp Card Reader”
xfsmIBRoot.3.1.3.8	(xfsmIBRoot.xfsTrap. xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	“Best Devices Incorporated”
xfsmIBRoot.3.1.3.9	(xfsmIBRoot.xfsTrap. xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	“1.10”
xfsmIBRoot.3.1.3.11	(xfsmIBRoot.xfsTrap. xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	“20/03/2003 15:40:53 -300”
xfsmIBRoot.3.1.3.12	(xfsmIBRoot.xfsTrap. xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	“1.23”
xfsmIBRoot.2.2.1.2.1. 3.Index	(xfsmIBRoot. xfsManagedService. xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry. xfsIDCStatusDevice .xfsIDCStatus ManagedServiceName)
	1 (WFS_STAT_DEVONLINE)
xfsmIBRoot.2.2.1.2.1. 2.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusNumberSubDevices .xfsIDCStatusManagedServiceName)
	0 (No sub device)
xfsmIBRoot.2.2.1.2.1. 4.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusMedia .xfsIDCStatusManagedServiceName)
	3 (xfsIDCMediaNotPresent)
xfsmIBRoot.2.2.1.2.1. 5.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusRetainBin .xfsIDCStatusManagedServiceName)
	2 (xfsIDCRetainBinOK)
xfsmIBRoot.2.2.1.2.1. 6.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusSecurity .xfsIDCStatusManagedServiceName)
	2 (xfsIDCSecurityNotSupported)
xfsmIBRoot.2.2.1.2.1. 7.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusCardRetained .xfsIDCStatusManagedServiceName)
	14 (14 cards retained)
xfsmIBRoot.2.2.1.2.1. 8.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusChipPower .xfsIDCStatusManagedServiceName)
	6 (xfsIDCChipNoCard)
xfsmIBRoot.2.2.1.2.1. 100.Index	(xfsmIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.x xfsIDCStatusExtraStatus .xfsIDCStatusManagedServiceName)
	"0"0' (No extra data)

4. Appendix A - IDC MIB sub-tree

The following paragraph contains the definition of the XFS IDC MIB sub-tree in ASN-1 format.

4.1 IDC MIB in SMIV2 and SMIV1 ASN-1 format

The following object contains the xfsIDC.MIB file in SMIV2 format.



SMIV2\xfsIDC.mib

The following object contains the xfsIDC.MIB file in SMIV1 format.



SMIV1\xfsIDC.mib

The following text is the content of xfsIDC.MIB in SMIV2 format.

```
-- *****
-- XFS MIB for IDC
-- Management Information Base for XFS IDC Device
--
-- The IDC Number is 2
-- The ASN.1 prefix to, and including the IDC is: 1.3.6.1.4.1.16213.2.2
--
-- *****
XFS-IDC-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        Integer32, OBJECT-TYPE, OBJECT-IDENTITY, NOTIFICATION-TYPE
            FROM SNMPv2-SMI
        DisplayString
            FROM SNMPv2-TC
        xfsIDC, xfsTrap, IxfsMIBDeviceStatus
            FROM XFSMIB;

-- *****
-- IDC #defines
-- *****
    IxfsIDCMediaStatus ::= INTEGER
    {
        xfsIDCMediaPresent(2),
        xfsIDCMediaNotPresent(3),
        xfsIDCMediaJammed(4),
        xfsIDCMediaNotSupported(5),
        xfsIDCMediaUnknown(6),
        xfsIDCMediaEntering(7),
        xfsIDCMediaLatched(8)
    }

    IxfsIDCRetainBinStatus ::= INTEGER
    {
        xfsIDCRetainBinOK(2),
        xfsIDCRetainBinNotSupported(3),
        xfsIDCRetainBinFull(4),
        xfsIDCRetainBinHigh(5)
    }

    IxfsIDCSecurityStatus ::= INTEGER
    {
        xfsIDCSecurityNotSupported(2),
        xfsIDCSecurityNotReady(3),
        xfsIDCSecurityOpen(4)
    }

    IxfsIDCChipPowerStatus ::= INTEGER
```

```

    {
        xfsIDCChipOnline(1),
        xfsIDCChipPoweredOff(2),
        xfsIDCChipBusy(3),
        xfsIDCChipNoDevice(4),
        xfsIDCChipHwError(5),
        xfsIDCChipNoCard(6),
        xfsIDCChipNotSupported(7),
        xfsIDCChipUnknown(8)
    }

-- *****
-- Version 1 of IDC MIB
--
-- The ASN.1 prefix to, and including the Version 1 of IDC is:
1.3.6.1.4.1.16213.2.2.1
--
-- *****
xfsIDCV1 OBJECT IDENTIFIER ::= { xfsIDC 1 }

xfsIDCInstances OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number that represents the number of IDC managed services."
    ::= { xfsIDCV1 1 }

-- *****
-- IDC Device Status Table
-- *****
xfsIDCStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIDCStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IDC status table."
    ::= { xfsIDCV1 2 }

xfsIDCStatusEntry OBJECT-TYPE
    SYNTAX XfsIDCStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IDC Device Status Table Entry."
    INDEX { xfsIDCStatusManagedServiceName }
    ::= { xfsIDCStatusTable 1 }

XfsIDCStatusEntry ::=
    SEQUENCE {
        xfsIDCStatusManagedServiceName
            DisplayString,
        xfsIDCStatusNumberSubDevices
            Integer32,
        xfsIDCStatusDevice
            IxfsMIBDeviceStatus,
        xfsIDCStatusMedia
            IxfsIDCMediaStatus,
        xfsIDCStatusRetainBin
            IxfsIDCRetainBinStatus,
        xfsIDCStatusSecurity
            IxfsIDCSecurityStatus,
        xfsIDCStatusCardRetained
            Integer32,
        xfsIDCStatusChipPower
            IxfsIDCChipPowerStatus,
        xfsIDCStatusExtraStatus
            OCTET STRING
    }

```

```

xfsIDCStatusManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIDCStatusEntry 1 }

xfsIDCStatusNumberSubDevices OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of sub devices supported by the IDC device."
    ::= { xfsIDCStatusEntry 2 }

xfsIDCStatusDevice OBJECT-TYPE
    SYNTAX IxfsMIBDeviceStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Device status."
    ::= { xfsIDCStatusEntry 3 }

xfsIDCStatusMedia OBJECT-TYPE
    SYNTAX IxfsIDCMediaStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Media Status.
        xfsIDCMediaPresent(2),
        xfsIDCMediaNotPresent(3),
        xfsIDCMediaJammed(4),
        xfsIDCMediaNotSupported(5),
        xfsIDCMediaUnknown(6),
        xfsIDCMediaEntering(7),
        xfsIDCMediaLatched(8)"
    ::= { xfsIDCStatusEntry 4 }

xfsIDCStatusRetainBin OBJECT-TYPE
    SYNTAX IxfsIDCRetainBinStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the ID card unit retain bin.
        xfsIDCRetainBinOK(2),
        xfsIDCRetainBinNotSupported(3),
        xfsIDCRetainBinFull(4),
        xfsIDCRetainBinHigh(5)"
    ::= { xfsIDCStatusEntry 5 }

xfsIDCStatusSecurity OBJECT-TYPE
    SYNTAX IxfsIDCSecurityStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the security unit.
        xfsIDCSecurityNotSupported(2),
        xfsIDCSecurityNotReady(3),
        xfsIDCSecurityOpen(4)"
    ::= { xfsIDCStatusEntry 6 }

xfsIDCStatusCardRetained OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of cards retained."

```

```

 ::= { xfsIDCStatusEntry 7 }

xfsIDCStatusChipPower OBJECT-TYPE
    SYNTAX IxfsIDCChipPowerStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Chip Power Status.
         xfsIDCChipOnline(1),
         xfsIDCChipPoweredOff(2),
         xfsIDCChipBusy(3),
         xfsIDCChipNoDevice(4),
         xfsIDCChipHwError(5),
         xfsIDCChipNoCard(6),
         xfsIDCChipNotSupported(7),
         xfsIDCChipUnknown(8)"
 ::= { xfsIDCStatusEntry 8 }

xfsIDCStatusExtraStatus OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Vendor dependent additional device status information."
 ::= { xfsIDCStatusEntry 100 }

-- *****
-- IDC Sub Device Status Table
--
-- Note that the IDC device does not currently have sub-devices. The
-- sub-device table is not required for this device and is shown as an
-- example for those devices that do support sub-devices.
--
-- Note, to ensure consistency across all MIB extensions OID 16213.2.2.1.3
-- must be reserved for the sub-device table.
-- *****

xfsIDCSubDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIDCSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IDC Sub-Device Status Table."
 ::= { xfsIDCV1 3 }

xfsIDCSubDeviceEntry OBJECT-TYPE
    SYNTAX XfsIDCSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IDC Sub-Device Status Table Entry."
    INDEX { xfsIDCSubDeviceManagedServiceName, xfsIDCSubDeviceIndex }
 ::= { xfsIDCSubDeviceTable 1 }

XfsIDCSubDeviceEntry ::=
    SEQUENCE {
        xfsIDCSubDeviceManagedServiceName
            DisplayString,
        xfsIDCSubDeviceIndex
            INTEGER
    }

-- As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
-- xfsIDCSubDeviceValue INTEGER }
xfsIDCSubDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current

```

```

DESCRIPTION
    "Instance identifier of the managed service."
    ::= { xfsIDCSubDeviceEntry 1 }

xfsIDCSubDeviceIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Index into the array of sub devices supported."
    ::= { xfsIDCSubDeviceEntry 2 }

-- As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
-- xfsIDCSubDeviceValue OBJECT-TYPE
-- SYNTAX INTEGER
-- ACCESS read-only
-- STATUS mandatory
-- DESCRIPTION "Returns the value of the sub device referenced by the index."
-- ::= {xfsIDCSubDeviceEntry 3}
-- *****
-- IDC Error Table
-- *****
xfsIDCErrorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIDCErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IDC Error Table."
    ::= { xfsIDCV1 4 }

xfsIDCErrorEntry OBJECT-TYPE
    SYNTAX XfsIDCErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IDC Error Table Entry."
    INDEX { xfsIDCErrorManagedServiceName, xfsIDCErrorCommandCode,
xfsIDCErrorResponseCode }
    ::= { xfsIDCErrorTable 1 }

XfsIDCErrorEntry ::=
    SEQUENCE {
        xfsIDCErrorManagedServiceName
            DisplayString,
        xfsIDCErrorCommandCode
            INTEGER,
        xfsIDCErrorResponseCode
            INTEGER,
        xfsIDCErrorCount
            Integer32
    }

xfsIDCErrorManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIDCErrorEntry 1 }

xfsIDCErrorCommandCode OBJECT-TYPE
    SYNTAX INTEGER (201..300)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The executable command code supported by the service
        provider associated with the error count of interest."
    ::= { xfsIDCErrorEntry 2 }

```

```

xfsIDCErrorResponseCode OBJECT-TYPE
    SYNTAX INTEGER (0..99 | 200..299)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The response code supported by service provider for the
        corresponding command code associated with the error count
        of interest."
    ::= { xfsIDCErrorEntry 3 }

xfsIDCErrorCount OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The counter value corresponding to the managed service,
        command code and response code."
    ::= { xfsIDCErrorEntry 4 }

-- *****
-- IDC Reset Table
-- *****
xfsIDCResetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIDCResetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Defines the set of MIB Variables for the IDC Reset Table."
    ::= { xfsIDCV1 5 }

xfsIDCResetEntry OBJECT-TYPE
    SYNTAX XfsIDCResetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IDC Reset Table Entry."
    INDEX { xfsIDCResetManagedServiceName }
    ::= { xfsIDCResetTable 1 }

XfsIDCResetEntry ::=
    SEQUENCE {
        xfsIDCResetManagedServiceName
            DisplayString,
        xfsIDCResetAll
            Integer32,
        xfsIDCResetTimestamp
            DisplayString
    }

xfsIDCResetManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIDCResetEntry 1 }

xfsIDCResetAll OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Returns all counter values for this managed service to
        zero when set to zero and returns zero when read."
    ::= { xfsIDCResetEntry 2 }

xfsIDCResetTimestamp OBJECT-TYPE
    SYNTAX DisplayString

```



```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Date and time the last reset of the counters was
    performed."
 ::= { xfsIDCResetEntry 3 }

-- *****
-- IDC Reset Device Table
-- *****
xfsIDCResetDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIDCResetDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IDC Reset Device Table."
    ::= { xfsIDCV1 6 }

xfsIDCResetDeviceEntry OBJECT-TYPE
    SYNTAX XfsIDCResetDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IDC Reset Device Table Entry."
    INDEX { xfsIDCResetDeviceManagedServiceName }
    ::= { xfsIDCResetDeviceTable 1 }

XfsIDCResetDeviceEntry ::=
    SEQUENCE {
        xfsIDCResetDeviceManagedServiceName
            DisplayString,
        xfsIDCResetDeviceAction
            INTEGER,
        xfsIDCResetDeviceMediaControl
            INTEGER,
        xfsIDCResetDeviceStatus
            INTEGER
    }

xfsIDCResetDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIDCResetDeviceEntry 1 }

xfsIDCResetDeviceAction OBJECT-TYPE
    SYNTAX INTEGER { executeReset(1) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Variable that initiates the device reset."
    ::= { xfsIDCResetDeviceEntry 2 }

xfsIDCResetDeviceMediaControl OBJECT-TYPE
    SYNTAX INTEGER
        {
            mediaDefault(1),
            mediaIn(2),
            mediaOut(3),
            mediaNoAction(4)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Variable that reports the media handling during the device reset."
    ::= { xfsIDCResetDeviceEntry 3 }

```

```

xfsIDCResetDeviceStatus OBJECT-TYPE
    SYNTAX INTEGER
    {
        resetIdle(1),
        resetInProgress(2)
    }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Variable that reports the progress of the device reset."
    ::= { xfsIDCResetDeviceEntry 4 }

xfsTrapV2 OBJECT-IDENTITY
    STATUS current
    DESCRIPTION
        "Root node for the converted TRAP-TYPES."
    ::= { xfsTrap 0 }

-- *****
-- Trap definitions
-- *****
xfsIDCDetailedDSCTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
              xfsCommonTrapManagedServiceClass,
              xfsCommonTrapManagedServiceClassName,
              xfsCommonTrapManagedServiceType,
              xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
              xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion,
              xfsCommonTrapEvent,
              xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsIDCStatusDevice,
              xfsIDCStatusNumberSubDevices, xfsIDCStatusMedia,
              xfsIDCStatusRetainBin, xfsIDCStatusSecurity,
              xfsIDCStatusCardRetained, xfsIDCStatusChipPower,
              xfsIDCStatusExtraStatus
            }
    STATUS current
    DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
    ::= { xfsTrapV2 102 }

xfsIDCResetDeviceCompleteTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
              xfsCommonTrapManagedServiceClass,
              xfsCommonTrapManagedServiceClassName,
              xfsCommonTrapManagedServiceType,
              xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
              xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion,
              xfsCommonTrapDate,
              xfsCommonTrapSPVersion, xfsIDCStatusDevice,
              xfsIDCStatusNumberSubDevices, xfsIDCStatusMedia,
              xfsIDCStatusRetainBin,
              xfsIDCStatusSecurity, xfsIDCStatusCardRetained,
              xfsIDCStatusChipPower, xfsIDCStatusExtraStatus }
    STATUS current
    DESCRIPTION
        "This trap indicates the Reset action has complete and reports the
        state of the device after the reset."
    ::= { xfsTrapV2 302 }

END

```

5. Appendix B - C-Header files

5.1 XFSMIBIDC.H



xfsmibdc.h

```

/*****
*
* xfsmibdc.h      WOSA/XFS - MIB IDC counters
*
*                Version 1.00  --  Jan 20, 2004
*
*****/

#ifndef __inc_xfsmibdc_h
#define __inc_xfsmibdc_h

#ifdef __cplusplus
extern "C" {
#endif

enum IxfsIDCMediaStatus
{
    xfsIDCMediaPresent           = 2,
    xfsIDCMediaNotPresent,
    xfsIDCMediaJammed,
    xfsIDCMediaNotSupported,
    xfsIDCMediaUnknown,
    xfsIDCMediaEntering,
    xfsIDCMediaLatched
} xfsIDCMediaStatus;

enum IxfsIDCRetainBinStatus
{
    xfsIDCRetainBinOK           = 2,
    xfsIDCRetainBinNotSupported,
    xfsIDCRetainBinFull,
    xfsIDCRetainBinHigh
} xfsIDCRetainBinStatus;

enum IxfsIDCSecurityStatus
{
    xfsIDCSecurityNotSupported   = 2,
    xfsIDCSecurityNotReady,
    xfsIDCSecurityOpen
} xfsIDCSecurityStatus;

enum IxfsIDCChipPowerStatus
{
    xfsIDCChipOnline           = 1,
    xfsIDCChipPoweredOff,
    xfsIDCChipBusy,
    xfsIDCChipNoDevice,
    xfsIDCChipHwError,
    xfsIDCChipNoCard,
    xfsIDCChipNotSupported,
    xfsIDCChipUnknown
} xfsIDCChipPowerStatus;

/*****

```

```

*
*           MIB Variables for the Status Table
*
*****/
#define      xfsIDCStatusManagedServiceName      (1)
#define      xfsIDCStatusNumberSubDevices        (2)
#define      xfsIDCStatusDevice                  (3)
#define      xfsIDCStatusMedia                   (4)
#define      xfsIDCStatusRetainBin               (5)
#define      xfsIDCStatusSecurity                (6)
#define      xfsIDCStatusCardRetained            (7)
#define      xfsIDCStatusChipPower               (8)
#define      xfsIDCStatusExtraStatus             (100)

/*****
*
*           MIB Variables for the Error Table
*
*****/
//Command codes and error codes correspond to the Service Provider definitions.

#ifdef __cplusplus
} /*extern "C"*/
#endif

#endif /* __inc_xfsmibidc__h */

```