CEN

WORKSHOP

AGREEMENT

CWA 16926-31

February 2017

ICS 35.240.40

English version

Extensions for Financial Services (XFS) interface specification – Release 3.30 – Part 31: XFS MIB Device Specific Definitions – Identification Card Device Class MIB 3.30

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, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, 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

© 2017 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.
Ref. No. CWA 16926-31:2017 D/E/F

Table of Contents

Foreword **3**

1. Introduction **7**

2. XFS IDC MIB variables **10**

2.1 XFS IDC STATUS TABLE 10
2.1.1 xfsIDCStatusTable: States *10*
2.2 XFS IDC SUB DEVICE TABLE 14
2.3 XFS IDC ERROR TABLE 14
2.4 XFS IDC RESET TABLE 15
2.5 XFS IDC RESET DEVICE TABLE 15
2.6 XFS IDC CAPABILITIES TABLE 16
2.6.1 xfsIDCCapabilitiesTable: Capabilities *16*

3. IDC Traps **23**

3.1 IDC DETAILED DEVICE STATUS CHANGE TRAP 23
3.1.1 IDC Detailed Device Status Change Trap Format *23*
3.1.2 IDC Detailed Device Status Change Trap: an example *26*
3.2 IDC SUB-DEVICE STATUS CHANGE TRAP 27
3.3 IDC RESET DEVICE COMPLETE TRAP 28
3.3.1 IDC Reset Device Complete Trap Format *28*
3.3.2 IDC Reset Device Complete: an example *30*

4. Appendix A - IDC MIB sub-tree **33**

4.1 IDC MIB IN SMIV2 AND SMIV1 ASN-1 FORMAT 33

5. Appendix B - C-Header files **49**

5.1 XFSMIBIDC.H 49

Foreword

This CWA is revision 3.30 of the XFS interface specification.

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties on March 19th 2015, the constitution of which was supported by CEN following the public call for participation made on 1998-06-24. 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 3.30.

A list of the individuals and organizations which supported the technical consensus represented by the CEN Workshop Agreement is available from the CEN/XFS Secretariat. The CEN XFS Workshop gathered suppliers as well as banks and other financial service companies.

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 and Scanning 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 Device Class Interface - Programmer's Reference

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

Part 16: Card Dispenser Device Class Interface - Programmer's Reference

Part 17: Barcode Reader Device Class Interface - Programmer's Reference

Part 18: Item Processing Module Device Class Interface- Programmer's Reference

Parts 19 - 28: Reserved for future use.

Parts 29 through 47 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 42: Reserved for future use.

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

Part 44: XFS MIB Application Management

Part 45: XFS MIB Device Specific Definitions - Card Dispenser Device Class

Part 46: XFS MIB Device Specific Definitions - Barcode Reader Device Class

Part 47: XFS MIB Device Specific Definitions - Item Processing Module Device Class

Parts 48 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Service Provider Interface (SPI) - Programmer's Reference

Part 62: Printer and Scanning Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 75: Card Dispenser Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 76: Barcode Reader Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

Part 77: Item Processing Module Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (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.cen.eu/work/areas/ict/ebusiness/pages/ws-xfs.aspx.

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 makes no warranty, express or implied, with respect to this document.

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

The final review/endorsement round for this CWA was started on 2015-01-16 and was successfully closed on 2015-03-19. The final text of this CWA was submitted to CEN for publication on 2015-06-19. The specification is continuously reviewed and commented in the CEN 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 3.30.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

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

Revision History:

1.0	January 20, 2004	Initial release of XFS MIB specification.
1.10	April 15, 2007	Update of the MIB to add support for a Detailed Status Trap, a Device Reset capability and the support of SMIv2.
3.10	December 14, 2010	Update of the MIB to add support for a Capabilities table and to align the MIB with XFS 3.10.
3.20	March 28, 2014	Update release to align the MIB with XFS 3.20.
3.30	February 6, 2017	Update release to align the MIB with XFS 3.30.

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:

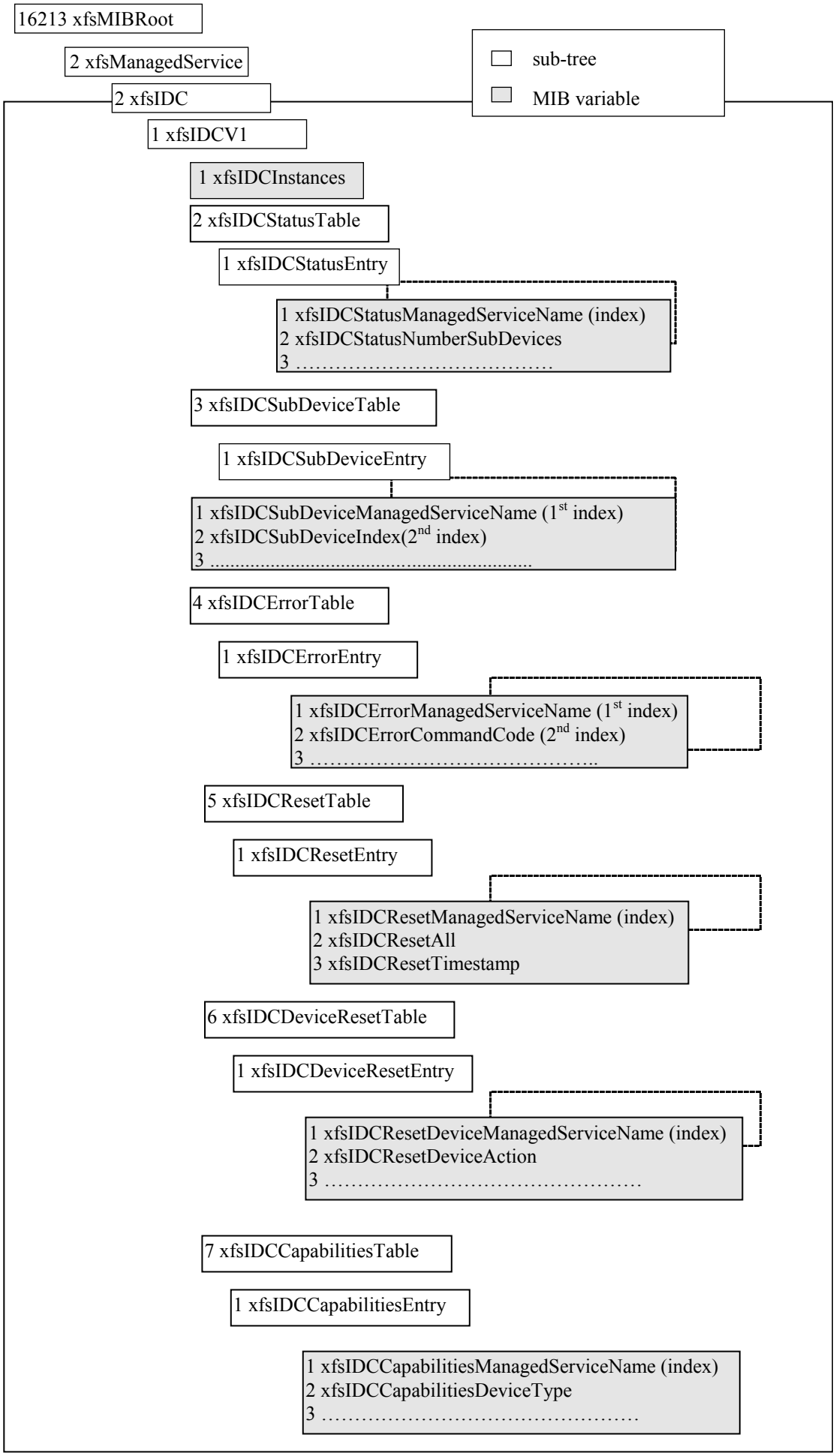
xfsMIBRoot

- 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.
- *xfsIDCCapabilitiesTable(7)* identifies the table for the DEP device capabilities 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.


16213 xfsMIBRoot
2 xfsManagedService
2 xfsIDC
1 xfsIDCV1
sub-tree
MIB variable
1 xfsIDCInstances
2 xfsIDCStatusTable
1 xfsIDCStatusEntry
1 xfsIDCStatusManagedServiceName (index)
2 xfsIDCStatusNumberSubDevices
3 ……
3 xfsIDCSubDeviceTable
1 xfsIDCSubDeviceEntry
1 xfsIDCSubDeviceManagedServiceName (1st index)
2 xfsIDCSubDeviceIndex(2nd index)
3 .....
4 xfsIDCErrorTable
1 xfsIDCErrorEntry
1 xfsIDCErrorManagedServiceName (1st index)
2 xfsIDCErrorCommandCode (2nd index)
3 ……..
5 xfsIDCResetTable
1 xfsIDCResetEntry
1 xfsIDCResetManagedServiceName (index)
2 xfsIDCResetAll
3 xfsIDCResetTimestamp
6 xfsIDCDeviceResetTable
1 xfsIDCDeviceResetEntry
1 xfsIDCResetDeviceManagedServiceName (index)
2 xfsIDCResetDeviceAction
3 ……
7 xfsIDCCapabilitiesTable
1 xfsIDCCapabilitiesEntry
1 xfsIDCCapabilitiesManagedServiceName (index)
2 xfsIDCCapabilitiesDeviceType
3 ……

Section 2 describes how the Status, Sub-Device, Error Reset, Reset Device and Capabilities 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(2)* 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	1
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 (zero) in the IDC device class.

xfsIDCStatusDevice (3)

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

Value	Meaning
xfsDevOnline(1)	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).

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.
xfsDevFraudAttempt(8)	The device is present but is inoperable because it has detected a fraud attempt.
xfsDevPotentialFraud(9)	The device has detected a potential fraud attempt and is capable of remaining in service.

xfsIDCStatusMedia (4)

It contains the media state. This status is independent of any media in the parking stations. 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. A card in a parking station is not considered to be present.
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 or contactless chip card reader).
xfsIDCMediaUnknown(6)	The media state cannot be determined with the device in its current state (e.g., the value of *fwDevice* 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 in a good state.
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.
xfsIDCRetainBinMissing(6)	The retain bin of the ID card unit is missing.

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 or is inoperable.
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. This value is returned for contactless chip card readers.
xfsIDCChipUnknown(8)	The state of the chip cannot be determined with the device in its current state.

xfsIDCStatusGuidanceCardUnit (9)

It contains the state of the card unit guidance light indicator.

Value	XFS Name	Meaning
0x00000000	WFS_IDC_GUIDANCE_NOT_AVAILABLE	The status is not available.
0x00000001	WFS_IDC_GUIDANCE_OFF	The light is turned off.
0x00000004	WFS_IDC_GUIDANCE_SLOW_FLASH	The light is blinking slowly.
0x00000008	WFS_IDC_GUIDANCE_MEDIUM_FLASH	The light is blinking medium frequency.
0x00000010	WFS_IDC_GUIDANCE_QUICK_FLASH	The light is blinking quickly.
0x00000080	WFS_IDC_GUIDANCE_CONTINUOUS	The light is turned on continuous (steady).
0x00000100	WFS_IDC_GUIDANCE_RED	The light is red.
0x00000200	WFS_IDC_GUIDANCE_GREEN	The light is green.
0x00000400	WFS_IDC_GUIDANCE_YELLOW	The light is yellow.
0x00000800	WFS_IDC_GUIDANCE_BLUE	The light is blue.
0x00001000	WFS_IDC_GUIDANCE_CYAN	The light is cyan.
0x00002000	WFS_IDC_GUIDANCE_MAGENTA	The light is magenta.
0x00004000	WFS_IDC_GUIDANCE_WHITE	The light is white.
0x00100000	WFS_IDC_GUIDANCE_ENTRY	The light is in the entry state.
0x00200000	WFS_IDC_GUIDANCE_EXIT	The light is the exit state.

xfsIDCStatusChipModule (10)

It contains the state of the chip card module reader. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCChipModuleOK(2)	The chip card module is in a good state.
xfsIDCChipModuleInop(3)	The chip card module is inoperable.
xfsIDCChipModuleUnknown(4)	The state of the chip card module cannot be determined.
xfsIDCChipModuleNotSupported(5)	Reporting the chip card module status is not supported.

xfsIDCStatusMagReadModule (11)

It contains the state of the magnetic card reader. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCMagModuleOK(2)	The magnetic card reading module is in a good state.
xfsIDCMagModuleInop(3)	The magnetic card reading module is inoperable.
xfsIDCMagModuleUnknown(4)	The state of the magnetic card reading module cannot be determined.
xfsIDCMagModuleNotSupported(5)	Reporting the magnetic card reading module status is not supported.

xfsIDCStatusMagWriteModule (12)

It contains the state of the magnetic card writer. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCMagModuleOK(2)	The magnetic card writing module is in a good state.
xfsIDCMagModuleInop(3)	The magnetic card writing module is inoperable.
xfsIDCMagModuleUnknown(4)	The state of the magnetic card writing module cannot be determined.
xfsIDCMagModuleNotSupported(5)	Reporting the magnetic card writing module status is not supported.

xfsIDCStatusFrontImageModule (13)

It contains the state of the front image reader. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCImgModuleOK(2)	The front image reading module is in a good state.
xfsIDCImgModuleInop(3)	The front image reading module is inoperable.
xfsIDCImgModuleUnknown(4)	The state of the front image reading module cannot be determined.
xfsIDCImgModuleNotSupported(5)	Reporting the front image reading module status is not supported.

xfsIDCStatusBackImageModule (14)

It contains the state of the back image reader. It is a numeric type field. Allowed values as follows:

Value	Meaning
xfsIDCImgModuleOK(2)	The back image reading module is in a good state.
xfsIDCImgModuleInop(3)	The back image reading module is inoperable.
xfsIDCImgModuleUnknown(4)	The state of the back image reading module cannot be determined.
xfsIDCImgModuleNotSupported(5)	Reporting the back image reading module status is not supported.

xfsIDCStatusDevicePosition (15)

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

Value	Meaning
xfsIDCDeviceInPosition(1)	The device is in its normal operating position, or is fixed in place and cannot be moved.
xfsIDCDeviceNotInPosition(2)	The device has been removed from its normal operating position.
xfsIDCDevicePosUnknown(3)	Due to a hardware error or other condition, the position of the device cannot be determined.
xfsIDCDevicePosNotSupp(4)	The physical device does not have the capability of detecting the position.

xfsIDCStatusPowerSaveRecoveryTime (16)

It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. This value is zero if either the power saving mode has not been activated or no power save control is supported. It is a numeric type field.

xfsIDCStatusParkingStationMedia (17)

It contains the states of the parking stations. It is an OCTET STRING. The information is returned as a series of ‘key=value’ strings where the ‘key’ is the parking position index and ‘value’ is the state of the parking position. Each string is null-terminated, with the final string terminating with two null characters. An empty list is indicated by two consecutive null characters. Valid parking position state values are as follows:

Value	XFS Name	Meaning
0x00000001	WFS_IDC_MEDIAPRESENT	Media is present in the parking station, and not jammed.
0x00000002	WFS_IDC_MEDIANOTPRESENT	Media is not present in the parking station.
0x00000003	WFS_IDC_MEDIAJAMMED	The parking station is jammed; operator intervention is required.
0x00000004	WFS_IDC_MEDIANOTSUPP	Reporting the media status in a parking station is not supported by the device.
0x00000005	WFS_IDC_MEDIAUNKNOWN	The media state cannot be determined.

xfsIDCStatusAntiFraudModule (18)

It contains the state of the anti-fraud module. It is a numeric type field. Allowed values are as follows:

Value	Meaning
xfsIDCAFMNotSupported(1)	No anti-fraud module is available.
xfsIDCAFMOK(2)	Anti-fraud module is in a good state and no foreign device is detected.
xfsIDCAFMInop(3)	Anti-fraud module is inoperable.
xfsIDCAFMDeviceDetected(4)	Anti-fraud module detected the presence of a foreign device.
xfsIDCAFMUnknown(5)	The state of the anti-fraud module cannot be determined.

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. An empty list is indicated by two consecutive 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(4)* 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 is 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 *xfsIDCResetTable(5)* contains the *xfsIDCResetAll* and *xfsIDCResetTimestamp* variables and is indexed by the single variable, *xfsIDCResetManagedServiceName*. When the *xfsIDCResetAll* 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 xfsIDCResetTable is defined as:

- *xfsIDCResetManagedServiceName(1)* which provides the index to the service in question. It is Display String field. The *xfsIDCResetManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table, e.g. “CardReader1”.
- *xfsIDCResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsIDCResetAll* 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 *xfsIDCResetAll* variable will return 0 (zero).
- *xfsIDCResetTimestamp(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 *xfsIDCResetAll* 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 *xfsIDCResetDeviceTable(6)* is indexed by the single variable, *xfsIDCResetDeviceManagedServiceName*. This table contains variables which monitor and control the execution of the reset request.

The *xfsIDCResetDeviceAction* 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 *xfsIDCResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request.

The *xfsIDCResetDeviceTable* 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 *xfsMIBRoot.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:

xfsMIBRoot.2.2.1.6.1.2.11.67.97.114.100.82.101.97.100.101.114.49

2.6 XFS IDC Capabilities Table

The *xfsIDCCapabilitiesTable(7)* groups the variables identifying device capabilities information. It is indexed through a single parameter, *xfsIDCCapabilitiesManagedServiceName*. All device capabilities 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.

xfsIDCCapabilitiesManagedServiceName 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 capabilities value of *xfsIDCCapabilitiesDeviceType(2)* for a device with managed service name equal to "CardReader1" is as follows:

Character	C	a	r	d	R	e	a	d	e	r	1
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.2.2.1.7.1.2.11.67.97.114.100.82.101.97.100.101.114.49

2.6.1 xfsIDCCapabilitiesTable: Capabilities

The first variable is common across all device classes, the other variables are device class specific.

xfsIDCCapabilitiesManagedServiceName (1)
Uniquely identifies the managed service.

xfsIDCCapabilitiesDeviceType (2)
Identifies the type of device as following numeric values.

Value	Meaning
xfsIDCTypeMotor(2)	The ID card unit is a motor driven card unit.
xfsIDCTypeSwipe(3)	The ID card unit is a swipe (pull-through) card unit .
xfsIDCTypeDip(4)	The ID card unit is a dip card unit. This DIP type is not capable of latching cards entered.
xfsIDCTypeContactless(5)	The ID card unit is a contactless card unit, i.e. no insertion of the card is required.
xfsIDCTypeLatchedDip(6)	The ID card unit is a latched dip card unit.
xfsIDCTypePermanent(7)	The ID card unit is dedicated to a permanently housed chip card (no user interaction is available with this type of card).
xfsIDCTypeIntelligentContactless(8)	The ID card unit is an intelligent contactless card unit.

xfsIDCCapabilitiesCompoundDevice (3)

It specifies whether the logical device is part of a compound physical device. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The logical device is a part of compound physical device.
False(2)	The logical device is not a part of compound physical device.

xfsIDCCapabilitiesReadTracks (4)

Specifies the tracks that can be read by the ID card unit as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0001	WFS_IDC_TRACK1	The ID card unit can access track 1.
0x0002	WFS_IDC_TRACK2	The ID card unit can access track 2.
0x0004	WFS_IDC_TRACK3	The ID card unit can access track 3.
0x8000	WFS_IDC_TRACK_WM	The ID card unit can access the Swedish Watermark track.
0x0080	WFS_IDC_FRONT_TRACK_1	The ID card unit can access the front track 1 (JIS II track).
0x0100	WFS_IDC_FRONTIMAGE	The ID card unit can access the front image of a card.
0x0200	WFS_IDC_BACKIMAGE	The ID card unit can access the back image of a card.
0x0400	WFS_IDC_TRACK1_JIS1	The ID card unit can access JIS I track 1.
0x0800	WFS_IDC_TRACK3_JIS1	The ID card unit can access JIS I track 3.
0x4000	WFS_IDC_DDI	The ID card unit can provide dynamic digital identification of the magnetic strip.

xfsIDCCapabilitiesWriteTracks (5)

Specifies the tracks that can be written by the ID card unit as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0001	WFS_IDC_TRACK1	The ID card unit can access track 1.
0x0002	WFS_IDC_TRACK2	The ID card unit can access track 2.
0x0004	WFS_IDC_TRACK3	The ID card unit can access track 3.
0x0080	WFS_IDC_FRONT_TRACK_1	The ID card unit can access the front track 1 (JIS II track).

xfsIDCCapabilitiesChipProtocols (6)

Specifies the chip card protocols that are supported by the Service Provider as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0000	WFS_IDC_NOTSUPP	The ID card unit cannot handle chip cards.
0x0001	WFS_IDC_CHIPT0	The ID card unit can handle the T=0 protocol.
0x0002	WFS_IDC_CHIPT1	The ID card unit can handle the T=1 protocol.
0x0004	WFS_IDC_CHIP_PROTOCOL_NOT_REQUIRED	The ID card unit is capable of communicating with a chip card without requiring the application to specify any

		protocol.
0x0008	WFS_IDC_CHIPTYPEA_PART3	The ID card unit can handle the ISO 14443 (Part3) Type A contactless chip card protocol.
0x0010	WFS_IDC_CHIPTYPEA_PART4	The ID card unit can handle the ISO 14443 (Part4) Type A contactless chip card protocol.
0x0020	WFS_IDC_CHIPTYPEB	The ID card unit can handle the ISO 14443 Type B contactless chip card protocol.
0x0040	WFS_IDC_CHIPNFC	The ID card unit can handle the ISO 18092 (106/212/424kbps) contactless chip card protocol.

xfsIDCCapabilitiesRetCards (7)

Specifies the maximum number of cards that can be held in the retain bin as numeric variable (zero if not available).

xfsIDCCapabilitiesSecurityType (8)

Specifies the type of security module used as one of the following values as numeric variable.

Value	Meaning
xfsIDCSecNotSupported(2)	Device has no security module.
xfsIDCSecMMBox(3)	Security module of device is MMBox.
xfsIDCSecCIM86(4)	Security module of device is CIM86.

xfsIDCCapabilitiesPowerOnOption (9)

Specifies the power-on capabilities of the device hardware as one of the following values (applicable only to motor driven ID card units) as numeric variable.

Value	Meaning
xfsIDCNoAction(2)	No power on actions are supported by the device
xfsIDCEject(3)	The card will be ejected on power-on (or off).
xfsIDCRetain(4)	The card will be retained on power-on (or off).
xfsIDCEjectThenRetain(5)	The card will be ejected for a specified time on power-on (or off), then retained if not taken. The time for which the card is ejected is vendor dependent.
xfsIDCReadPosition(6)	The card will be moved into the read position on power-on (or off).

xfsIDCCapabilitiesPowerOffOption (10)

Specifies the power-off capabilities of the device hardware as one of the following values (applicable only to motor driven ID card units) as numeric variable. Allowed values are the same as variable *xfsIDCCapabilitiesPowerOnOption (9)*.

xfsIDCCapabilitiesProgrammableFlux (11)

Specifies whether the flux sensor is programmable or not in TruthValue format. Allowed values are:

Value	Meaning
True(1)	The device has the capability.
False(2)	The device does not have the capability.

xfsIDCCapabilitiesReadWriteAfterEject (12)

Specifies whether the card can be read or written to after ejected to the exit slot in TruthValue format. Allowed values are:

Value	Meaning
True(1)	The device has the capability.

False(2) The device does not have the capability.

xfsIDCCapabilitiesWriteMode (13)

Specifies the write capabilities, with respect to whether the device can write low coercivity (loco) and / or high coercivity (hico) magnetic stripes, as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0000	WFS_IDC_NOTSUPP	Does not support writing of magnetic stripes.
0x0002	WFS_IDC_LOCO	The ID card unit supports writing of loco magnetic stripes.
0x0004	WFS_IDC_HICO	The ID card unit supports writing of hico magnetic stripes.
0x0008	WFS_IDC_AUTO	The ID card unit is capable of automatically determining whether loco or hico magnetic stripes should be written.

xfsIDCCapabilitiesChipPower (14)

Specifies the capabilities of the ID card unit (in relation to the user or permanent chip controlled by the service), for chip power management as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0000	WFS_IDC_NOTSUPP	The ID card unit cannot handle chip power management.
0x0002	WFS_IDC_CHIPPOWERCOLD	The ID card unit can power on the chip and reset it (Cold Reset).
0x0004	WFS_IDC_CHIPPOWERWARM	The ID card unit can reset the chip (Warm Reset).
0x0008	WFS_IDC_CHIPPOWEROFF	The ID card unit can power off the chip.

xfsIDCCapabilitiesDipMode (15)

Specifies whether data track data is read on entry or exit from the dip card unit as one of the following flags as a numeric variable.

Value	Meaning
xfsIDCNotSupported (1)	The ID card unit is not a dip type.
xfsIDCDipExit (3)	The dip ID card unit reads card track data on exit only.
xfsIDCDipEntry (5)	The dip ID card unit reads card track data on entry only.
xfsIDCDipEntryExit (9)	The dip ID card unit reads card track data both on entry and exit.
xfsIDCDipUnknown (2)	Unknown whether track data is read on entry or exit.

xfsIDCCapabilitiesMemoryChipProtocols (16)

Specifies the memory card protocols that are supported by the Service Provider as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0000	WFS_IDC_NOTSUPP	The device does not support any memory card protocols.
0x0001	WFS_IDC_MEM_SIEMENS4442	The device supports the Siemens 4442 Card Protocol (also supported by the Gemplus GPM2K card).
0x0002	WFS_IDC_MEM_GPM896	The device supports the Gemplus GPM 896 Card Protocol.

xfsIDCCapabilitiesGuidanceCardUnit (17)

This integer variable specifies the guidance light capability of the card unit. Possible states are reported as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning

0x00000000	WFS_IDC_GUIDANCE_NOT_AVAILABLE	There is no guidance control available at this position.
0x00000001	WFS_IDC_GUIDANCE_OFF	The light can be off.
0x00000004	WFS_IDC_GUIDANCE_SLOW_FLASH	The light can blink slowly.
0x00000008	WFS_IDC_GUIDANCE_MEDIUM_FLASH	The light can blink medium frequency.
0x00000010	WFS_IDC_GUIDANCE_QUICK_FLASH	The light can blink quickly.
0x00000080	WFS_IDC_GUIDANCE_CONTINUOUS	The light can be continuous (steady).
0x00000100	WFS_IDC_GUIDANCE_RED	The light can be red.
0x00000200	WFS_IDC_GUIDANCE_GREEN	The light can be green.
0x00000400	WFS_IDC_GUIDANCE_YELLOW	The light can be yellow.
0x00000800	WFS_IDC_GUIDANCE_BLUE	The light can be blue.
0x00001000	WFS_IDC_GUIDANCE_CYAN	The light can be cyan.
0x00002000	WFS_IDC_GUIDANCE_MAGENTA	The light can be magenta.
0x00004000	WFS_IDC_GUIDANCE_WHITE	The light can be white.
0x00100000	WFS_IDC_GUIDANCE_ENTRY	The light is in the entry state.
0x00200000	WFS_IDC_GUIDANCE_EXIT	The light is the exit state.

xfsIDCCapabilitiesEjectPosition (18)

Specifies the target position that is supported for the eject operation as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0001	WFS_IDC_EXITPOSITION	The device can eject a card to the exit position, from which the user can remove it.
0x0002	WFS_IDC_TRANSPORTPOSITION	The device can eject a card to the transport just behind the exit position, from which the user cannot remove it.

xfsIDCCapabilitiesPowerSaveControl (19)

Specifies the capability of the power saving control. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	Power saving is supported.
False(2)	Power saving is not supported.

xfsIDCCapabilitiesParkingStations (20)

Specifies the number of supported parking stations. If a zero value is specified there is no parking station supported. It is a numeric variable.

xfsIDCCapabilitiesAntiFraudModule (21)

Specifies whether the anti-fraud module is available in a TruthValue variable as follows:

Value	Meaning
True (1)	The device has an anti-fraud module.
False (2)	The device does not have an anti-fraud module.

xfsIDCCapabilitiesSynchronizableCommands (22)

This holds the list of execute command IDs that can be synchronized. It is an OCTET STRING with each command ID separated by a NULL terminator, while the field itself is terminated by a double NULL

terminator. For example, for a device that can synchronize more than one command ID, this value will be as follows where <null> is null terminator:

$commandID_1$<null>$commandID_2$<null>……$commandID_n$<null><null>

If there are no command IDs that can be synchronized then the value is as follows where <null> is null terminator:

<null><null>

Possible values for each command ID are:

Value	XFS Name	Meaning
201	WFS_CMD_IDC_READ_TRACK	The WFS_CMD_IDC_READ_TRACK command can be synchronized.
202	WFS_CMD_IDC_WRITE_TRACK	The WFS_CMD_IDC_WRITE_TRACK command can be synchronized.
203	WFS_CMD_IDC_EJECT_CARD	The WFS_CMD_IDC_EJECT_CARD command can be synchronized.
204	WFS_CMD_IDC_RETAIN_CARD	The WFS_CMD_IDC_RETAIN_CARD command can be synchronized.
205	WFS_CMD_IDC_RESET_COUNT	The WFS_CMD_IDC_RESET_COUNT command can be synchronized.
206	WFS_CMD_IDC_SETKEY	The WFS_CMD_IDC_SETKEY command can be synchronized.
207	WFS_CMD_IDC_READ_RAW_DATA	The WFS_CMD_IDC_READ_RAW_DATA command can be synchronized.
208	WFS_CMD_IDC_WRITE_RAW_DATA	The WFS_CMD_IDC_WRITE_RAW_DATA command can be synchronized.
209	WFS_CMD_IDC_CHIP_IO	The WFS_CMD_IDC_CHIP_IO command can be synchronized.
210	WFS_CMD_IDC_RESET	The WFS_CMD_IDC_RESET command can be synchronized.
211	WFS_CMD_IDC_CHIP_POWER	The WFS_CMD_IDC_CHIP_POWER command can be synchronized.
212	WFS_CMD_IDC_PARSE_DATA	The WFS_CMD_IDC_PARSE_DATA command can be synchronized.
213	WFS_CMD_IDC_SET_GUIDANCE_LIGHT	The WFS_CMD_IDC_SET_GUIDANCE_LIGHT command can be synchronized.
214	WFS_CMD_IDC_POWER_SAVE_CONTROL	The WFS_CMD_IDC_POWER_SAVE_CONTROL command can be synchronized.
215	WFS_CMD_IDC_PARK_CARD	The WFS_CMD_IDC_PARK_CARD command can be synchronized.
216	WFS_CMD_IDC_EMVCLESS_CONFIGURE	The WFS_CMD_IDC_EMVCLESS_CONFIGURE command can be synchronized.
217	WFS_CMD_IDC_EMVCLESS_PERFORM_TRANSACTION	The WFS_CMD_IDC_EMVCLESS_PERFORM_TRANSACTION command can be synchronized.

218	WFS_CMD_IDC_EMVCLESS_ISSUERUPDATE	The WFS_CMD_IDC_EMVCLESS_ISSUER UPDATE command can be synchronized.

xfsIDCCapabilitiesExtraCapability (100)

It contains the vendor dependent additional device capability 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. 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. 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.

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\<Managed*ServiceName*>\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.**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.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusGuidanceCardUnit.**xfsIDCStatusManagedServiceName (21)

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

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusChipModule.**xfsIDCStatusManagedServiceName (22)

It contains the state of the chip card module reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagReadModule.**xfsIDCStatusManagedServiceName (23)

It contains the state of the magnetic card reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagWriteModule.**xfsIDCStatusManagedServiceName (24)

It contains the state of the magnetic card writer. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusFrontImageModule.**xfsIDCStatusManagedServiceName (25)

It contains the state of the front image reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusBackImageModule.**xfsIDCStatusManagedServiceName (26)

It contains the state of the back image reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusDevicePosition.**xfsIDCStatusManagedServiceName (27)

It contains the device position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusPowerSaveRecoveryTime.**xfsIDCStatusManagedServiceName (28)

It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusParkingStationMedia.**xfsIDCStatusManagedServiceName (29)

It contains the states of the parking stations. It is an OCTET STRING. The information is returned as a series of ‘key=value’ strings where the ‘key’ is the parking position index and ‘value’ is the state of the parking station. Each string is null-terminated, with the final string terminating with two null characters. An empty list is indicated by two consecutive null characters.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusAntiFraudModule.**xfsIDCStatusManagedServiceName (30)

It contains the state of the anti-fraud module. It is a numeric type field.

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)
xfsMIBRoot.2.2.1.2.1.**9**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusGuidanceCardUnit**.xfsIDCStatusManagedServiceName)
	1 (value corresponding to WFS_IDC_GUIDANCE_OFF)
xfsMIBRoot.2.2.1.2.1.**10**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusChipModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCChipModuleOK)
xfsMIBRoot.2.2.1.2.1.**11**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagReadModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCMagModuleOK)
xfsMIBRoot.2.2.1.2.1.**12**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagWriteModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCMagModuleOK)
xfsMIBRoot.2.2.1.2.1.**13**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusFrontImageModule**.xfsIDCStatusManagedServiceName)
	5 (xfsIDCImgModuleNotSupported)
xfsMIBRoot.2.2.1.2.1.**14**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusBackImageModule**.xfsIDCStatusManagedServiceName)
	5 (xfsIDCImgModuleNotSupported)
xfsMIBRoot.2.2.1.2.1.**15**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusDevicePosition**.xfsIDCStatusManagedServiceName)
	1 (xfsIDCDeviceInPosition)
xfsMIBRoot.2.2.1.2.1.**16**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusPowerSaveRecoveryTime**.xfsIDCStatusManagedServiceName)
	0 (Power save control is not supported)
xfsMIBRoot.2.2.1.2.1.**17**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusParkingStationMedia**.xfsIDCStatusManagedServiceName)
	'\0"\0' (No parking stations supported)
xfsMIBRoot.2.2.1.2.1.**18**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusAntiFraudModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCAFMOK)

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 an 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\<Managed*ServiceName*>\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.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusGuidanceCardUnit.**xfsIDCStatusManagedServiceName (21)
It contains the state of the card unit guidance. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusChipModule.**xfsIDCStatusManagedServiceName (22)

It contains the state of the chip card module reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagReadModule.**xfsIDCStatusManagedServiceName (23)

It contains the state of the magnetic card reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagWriteModule.**xfsIDCStatusManagedServiceName (24)

It contains the state of the magnetic card writer. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusFrontImageModule.**xfsIDCStatusManagedServiceName (25)

It contains the state of the front image reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusBackImageModule.**xfsIDCStatusManagedServiceName (26)

It contains the state of the back image reader. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusDevicePosition.**xfsIDCStatusManagedServiceName (27)

It contains the device position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusPowerSaveRecoveryTime.**xfsIDCStatusManagedServiceName (28)

It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusParkingStationMedia.**xfsIDCStatusManagedServiceName (29)

It contains the states of the parking stations. It is an OCTET STRING. The information is returned as a series of ‘key=value’ strings where the ‘key’ is the parking position index and ‘value’ is state of the parking station. Each string is null-terminated, with the final string terminating with two null characters. An empty list is indicated by two consecutive null characters.

xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusAntiFraudModule.**xfsIDCStatusManagedServiceName (30)

It contains the state of the anti-fraud module. It is a numeric type field.

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 the result of a request to reset the device from the remote management station. The device in question has a managed service name “CardReader1”.

xfsMIBRoot.3.1.3.13	(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.xfsCommonTrapManagedServiceClass)
	2 (WFS_SERVICE_CLASS_IDC)
xfsMIBRoot.3.1.3.4	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClas

	sName)
	“IDC”
xfsMIBRoot.3.1.3.5	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceTyp e)
	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.xfsCommonTrapPhysicalDeviceNam e)
	“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**.xfsIDCStatusManagedServiceName)
	1 (WFS_STAT_DEVONLINE)
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)
	3 (xfsIDCMediaNotPresent)
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)
	6 (xfsIDCChipNoCard)
xfsMIBRoot.2.2.1.2.1.**100**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusExtraStatus**.xfsIDCStatusManagedServiceName)
	'\0"\0' (No extra data)
xfsMIBRoot.2.2.1.2.1.**9**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusGuidanceCardUnit**.xfsIDCStatusManagedServiceName)
	1 (value corresponding to WFS_IDC_GUIDANCE_OFF)
xfsMIBRoot.2.2.1.2.1.**10**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusChipModule**.xfsIDCStatusManagedServiceName)

	2 (xfsIDCChipModuleOK)
xfsMIBRoot.2.2.1.2.1.**11**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagReadModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCMagModuleOK)
xfsMIBRoot.2.2.1.2.1.**12**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusMagWriteModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCMagModuleOK)
xfsMIBRoot.2.2.1.2.1.**13**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusFrontImageModule**.xfsIDCStatusManagedServiceName)
	5 (xfsIDCImgModuleNotSupported)
xfsMIBRoot.2.2.1.2.1.**14**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusBackImageModule**.xfsIDCStatusManagedServiceName)
	5 (xfsIDCImgModuleNotSupported)
xfsMIBRoot.2.2.1.2.1.**15**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusDevicePosition**.xfsIDCStatusManagedServiceName)
	1 (xfsIDCDeviceInPosition)
xfsMIBRoot.2.2.1.2.1.**16**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusPowerSaveRecoveryTime**.xfsIDCStatusManagedServiceName)
	0 (Power save control is not supported)
xfsMIBRoot.2.2.1.2.1.**17**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusParkingStationMedia**.xfsIDCStatusManagedServiceName)
	'\0"\0' (No parking stations supported)
xfsMIBRoot.2.2.1.2.1.**18**.Index	(xfsMIBRoot.xfsManagedService.xfsIDC.xfsIDCV1.xfsIDCStatusTable.xfsIDCStatusEntry.**xfsIDCStatusAntiFraudModule**.xfsIDCStatusManagedServiceName)
	2 (xfsIDCAFMOK)

4. Appendix A - IDC MIB sub-tree

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

SMIv1_xfsIDC.mib SMIv2_xfsIDC.mib

4.1 IDC MIB in SMIv2 and SMIv1 ASN-1 format

The following text is the content of xfsIDC.MIB in SMIv2 format.

```
-- ******************************************************************
-- XFS 3.30 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, TruthValue
            FROM SNMPv2-TC
         xfsIDC, xfsTrap, IxfsMIBDeviceStatus
            FROM XFSMIB;


--
-- Type definitions
--

-- ******************************************************************
-- 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),
         xfsIDCRetainBinMissing(6)
         }

      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)
      }

   IxfsIDCChipModuleStatus ::= INTEGER
      {
      xfsIDCChipModuleOK(2),
      xfsIDCChipModuleInop(3),
      xfsIDCChipModuleUnknown(4),
      xfsIDCChipModuleNotSupported(5)
      }

   IxfsIDCMagModuleStatus ::= INTEGER
      {
      xfsIDCMagModuleOK(2),
      xfsIDCMagModuleInop(3),
      xfsIDCMagModuleUnknown(4),
      xfsIDCMagModuleNotSupported(5)
      }

   IxfsIDCImageModuleStatus ::= INTEGER
      {
      xfsIDCImageModuleOK(2),
      xfsIDCImageModuleInop(3),
      xfsIDCImageModuleUnknown(4),
      xfsIDCImageModuleNotSupported(5)
      }

   IxfsIDCDevicePositionStatus ::= INTEGER
      {
      xfsIDCDeviceInPosition(1),
      xfsIDCDeviceNotInPosition(2),
      xfsIDCDevicePosUnknown(3),
      xfsIDCDevicePosNotSupported(4)
      }

   IxfsIDCAntiFraudModuleStatus ::= INTEGER
      {
      xfsIDCAFMNotSupported(1),
      xfsIDCAFMOK(2),
      xfsIDCAFMInop(3),
      xfsIDCAFMDeviceDetected(4),
      xfsIDCAFMUnknown(5)
      }

   IxfsIDCCapabilitiesDeviceType ::= INTEGER
      {
      xfsIDCTypeMotor(2),
      xfsIDCTypeSwipe(3),
      xfsIDCTypeDip(4),
      xfsIDCTypeContactless(5),
      xfsIDCTypeLatchedDip(6),
      xfsIDCTypePermanent(7),
      xfsIDCTypeIntelligentContactless(8)
      }

   IxfsIDCCapabilitiesSecurityType ::= INTEGER
      {
      xfsIDCSecNotSupported(2),
      xfsIDCSecMMBox(3),
      xfsIDCSecCIM86(4)
      }

   IxfsIDCCapabilitiesPowerOptions ::= INTEGER
      {
      xfsIDCNoAction(2),
      xfsIDCEject(3),
      xfsIDCRetain(4),
      xfsIDCEjectThenRetain(5),
      xfsIDCReadPosition(6)
      }

   IxfsIDCCapabilitiesDIPMode ::= INTEGER
      {
```

```
        xfsIDCNotSupported(1),
        xfsIDCDipUnknown(2),
        xfsIDCDipExit(3),
        xfsIDCDipEntry(5),
        xfsIDCDipEntryExit(9)
        }



--
-- Node definitions
--

-- ******************************************************************
-- 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
--
-- ******************************************************************
     -- 1.3.6.1.4.1.16213.2.2.1
     xfsIDCV1 OBJECT IDENTIFIER ::= { xfsIDC 1 }


     -- 1.3.6.1.4.1.16213.2.2.1.1
     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
-- ********************************************************************
     -- 1.3.6.1.4.1.16213.2.2.1.2
     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 }


     -- 1.3.6.1.4.1.16213.2.2.1.2.1
     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,
      xfsIDCStatusGuidanceCardUnit
         Integer32,
      xfsIDCStatusChipModule
         IxfsIDCChipModuleStatus,
      xfsIDCStatusMagReadModule
         IxfsIDCMagModuleStatus,
      xfsIDCStatusMagWriteModule
         IxfsIDCMagModuleStatus,
      xfsIDCStatusFrontImageModule
         IxfsIDCImageModuleStatus,
      xfsIDCStatusBackImageModule
         IxfsIDCImageModuleStatus,
      xfsIDCStatusDevicePosition
         IxfsIDCDevicePositionStatus,
      xfsIDCStatusPowerSaveRecoveryTime
         Integer32,
      xfsIDCStatusParkingStationMedia
         OCTET STRING,
      xfsIDCStatusAntiFraudModule
         IxfsIDCAntiFraudModuleStatus,
      xfsIDCStatusExtraStatus
         OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.2.1.2.1.1
xfsIDCStatusManagedServiceName OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Instance identifier of the managed service."
   ::= { xfsIDCStatusEntry 1 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.2
xfsIDCStatusNumberSubDevices OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Number of sub devices supported by the IDC device."
   ::= { xfsIDCStatusEntry 2 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.3
xfsIDCStatusDevice OBJECT-TYPE
   SYNTAX IxfsMIBDeviceStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Device status."
   ::= { xfsIDCStatusEntry 3 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.4
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 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.5
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 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.6
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 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.7
xfsIDCStatusCardRetained OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Number of cards retained."
   ::= { xfsIDCStatusEntry 7 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.8
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 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.9
xfsIDCStatusGuidanceCardUnit OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Contains the state of the card unit guidance light indicator."
   ::= { xfsIDCStatusEntry 9 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.10
xfsIDCStatusChipModule OBJECT-TYPE
   SYNTAX IxfsIDCChipModuleStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Contains the state of the chip card module reader.
      xfsIDCChipModuleOK(2)
      xfsIDCChipModuleInop(3)
      xfsIDCChipModuleUnknown(4)
      xfsIDCChipModuleNotSupported(5)"
   ::= { xfsIDCStatusEntry 10 }
```

```
-- 1.3.6.1.4.1.16213.2.2.1.2.1.11
xfsIDCStatusMagReadModule OBJECT-TYPE
   SYNTAX IxfsIDCMagModuleStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Contains the state of the magnetic card reader.
      xfsIDCMagModuleOK(2)
      xfsIDCMagModuleInop(3)
      xfsIDCMagModuleUnknown(4)
      xfsIDCMagModuleNotSupported(5)"
   ::= { xfsIDCStatusEntry 11 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.12
xfsIDCStatusMagWriteModule OBJECT-TYPE
   SYNTAX IxfsIDCMagModuleStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Contains the state of the magnetic card writer.
      xfsIDCMagModuleOK(2)
      xfsIDCMagModuleInop(3)
      xfsIDCMagModuleUnknown(4)
      xfsIDCMagModuleNotSupported(5)"
   ::= { xfsIDCStatusEntry 12 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.13
xfsIDCStatusFrontImageModule OBJECT-TYPE
   SYNTAX IxfsIDCImageModuleStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Contains the state of the front image reader.
      xfsIDCImgModuleOK(2)
      xfsIDCImgModuleInop(3)
      xfsIDCImgModuleUnknown(4)
      xfsIDCImgModuleNotSupported(5)"
   ::= { xfsIDCStatusEntry 13 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.14
xfsIDCStatusBackImageModule OBJECT-TYPE
   SYNTAX IxfsIDCImageModuleStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Contains the state of the back image reader.
      xfsIDCImgModuleOK(2)
      xfsIDCImgModuleInop(3)
      xfsIDCImgModuleUnknown(4)
      xfsIDCImgModuleNotSupported(5)"
   ::= { xfsIDCStatusEntry 14 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.15
xfsIDCStatusDevicePosition OBJECT-TYPE
   SYNTAX IxfsIDCDevicePositionStatus
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Specifies the device position.
      xfsIDCDeviceInPosition(1),
      xfsIDCDeviceNotInPosition(2),
      xfsIDCDevicePosUnknown(3),
      xfsIDCDevicePosNotSupported(4)."
   ::= { xfsIDCStatusEntry 15 }


-- 1.3.6.1.4.1.16213.2.2.1.2.1.16
xfsIDCStatusPowerSaveRecoveryTime OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
```

```
      STATUS current
      DESCRIPTION
         "Specifies the actual number of seconds required by the
         device to resume its normal operational state from the
         current power saving mode. This value is zero if either the
         power saving mode has not been activated or no power save
         control is supported."
      ::= { xfsIDCStatusEntry 16 }


   -- 1.3.6.1.4.1.16213.2.2.1.2.1.17
   xfsIDCStatusParkingStationMedia OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
         "It contains the states of the parking stations."
      ::= { xfsIDCStatusEntry 17 }


   -- 1.3.6.1.4.1.16213.2.2.1.2.1.18
   xfsIDCStatusAntiFraudModule OBJECT-TYPE
      SYNTAX IxfsIDCAntiFraudModuleStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
      "Specifies the state of the anti-fraud module.
       xfsIDCAFMNotSupported(1),
       xfsIDCAFMOK(2),
       xfsIDCAFMInop(3),
       xfsIDCAFMDeviceDetected(4),
       xfsIDCAFMUnknown(5)."
      ::= { xfsIDCStatusEntry 18 }


   -- 1.3.6.1.4.1.16213.2.2.1.2.1.100
   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.
-- ********************************************************************
   -- 1.3.6.1.4.1.16213.2.2.1.3
   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 }


   -- 1.3.6.1.4.1.16213.2.2.1.3.1
   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 }
     -- 1.3.6.1.4.1.16213.2.2.1.3.1.1
     xfsIDCSubDeviceManagedServiceName OBJECT-TYPE
        SYNTAX DisplayString
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "Instance identifier of the managed service."
        ::= { xfsIDCSubDeviceEntry 1 }


     -- 1.3.6.1.4.1.16213.2.2.1.3.1.2
     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
-- ********************************************************************
     -- 1.3.6.1.4.1.16213.2.2.1.4
     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 }


     -- 1.3.6.1.4.1.16213.2.2.1.4.1
     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
        }

     -- 1.3.6.1.4.1.16213.2.2.1.4.1.1
     xfsIDCErrorManagedServiceName OBJECT-TYPE
        SYNTAX DisplayString
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "Instance identifier of the managed service."
        ::= { xfsIDCErrorEntry 1 }


     -- 1.3.6.1.4.1.16213.2.2.1.4.1.2
     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 }


     -- 1.3.6.1.4.1.16213.2.2.1.4.1.3
     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 }


     -- 1.3.6.1.4.1.16213.2.2.1.4.1.4
     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
-- ***********************************************************************
     -- 1.3.6.1.4.1.16213.2.2.1.5
     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 }


     -- 1.3.6.1.4.1.16213.2.2.1.5.1
     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
       }

   -- 1.3.6.1.4.1.16213.2.2.1.5.1.1
   xfsIDCResetManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
         "Instance identifier of the managed service."
      ::= { xfsIDCResetEntry 1 }


   -- 1.3.6.1.4.1.16213.2.2.1.5.1.2
   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 }


   -- 1.3.6.1.4.1.16213.2.2.1.5.1.3
   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
-- ********************************************************************
   -- 1.3.6.1.4.1.16213.2.2.1.6
   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 }


   -- 1.3.6.1.4.1.16213.2.2.1.6.1
   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
       }

   -- 1.3.6.1.4.1.16213.2.2.1.6.1.1
```

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


      -- 1.3.6.1.4.1.16213.2.2.1.6.1.2
      xfsIDCResetDeviceAction OBJECT-TYPE
         SYNTAX INTEGER { executeReset(1) }
         MAX-ACCESS read-write
         STATUS current
         DESCRIPTION
            "Variable that initiates the device reset."
         ::= { xfsIDCResetDeviceEntry 2 }


      -- 1.3.6.1.4.1.16213.2.2.1.6.1.3
      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 }


      -- 1.3.6.1.4.1.16213.2.2.1.6.1.4
      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 }


-- ***********************************************************************
-- IDC Device Capabilities Table
-- ***********************************************************************
      -- 1.3.6.1.4.1.16213.2.2.1.7
      xfsIDCCapabilitiesTable OBJECT-TYPE
         SYNTAX SEQUENCE OF XfsIDCCapabilitiesEntry
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
            "Define the set of MIB Variables for the IDC capabilities table."
         ::= { xfsIDCV1 7 }


      -- 1.3.6.1.4.1.16213.2.2.1.7.1
      xfsIDCCapabilitiesEntry OBJECT-TYPE
         SYNTAX XfsIDCCapabilitiesEntry
         MAX-ACCESS not-accessible
         STATUS current
         DESCRIPTION
            "IDC Device Capabilities Table Entry."
         INDEX { xfsIDCCapabilitiesManagedServiceName }
         ::= { xfsIDCCapabilitiesTable 1 }


      XfsIDCCapabilitiesEntry ::=
         SEQUENCE {
```

```
      xfsIDCCapabilitiesManagedServiceName
         DisplayString,
      xfsIDCCapabilitiesDeviceType
         IxfsIDCCapabilitiesDeviceType,
      xfsIDCCapabilitiesCompoundDevice
         TruthValue,
      xfsIDCCapabilitiesReadTracks
         Integer32,
      xfsIDCCapabilitiesWriteTracks
         Integer32,
      xfsIDCCapabilitiesChipProtocols
         Integer32,
      xfsIDCCapabilitiesRetCards
         Integer32,
      xfsIDCCapabilitiesSecurityType
         IxfsIDCCapabilitiesSecurityType,
      xfsIDCCapabilitiesPowerOnOption
         IxfsIDCCapabilitiesPowerOptions,
      xfsIDCCapabilitiesPowerOffOption
         IxfsIDCCapabilitiesPowerOptions,
      xfsIDCCapabilitiesProgrammableFlux
         TruthValue,
      xfsIDCCapabilitiesReadWriteAfterEject
         TruthValue,
      xfsIDCCapabilitiesWriteMode
         Integer32,
      xfsIDCCapabilitiesChipPower
         Integer32,
      xfsIDCCapabilitiesDipMode
         IxfsIDCCapabilitiesDIPMode,
      xfsIDCCapabilitiesMemoryChipProtocols
         Integer32,
      xfsIDCCapabilitiesGuidanceCardUnit
         Integer32,
      xfsIDCCapabilitiesEjectPosition
         Integer32,
      xfsIDCCapabilitiesPowerSaveControl
         TruthValue,
      xfsIDCCapabilitiesParkingStations
         Integer32,
      xfsIDCCapabilitiesAntiFraudModule
         TruthValue,
      xfsIDCCapabilitiesSynchronizableCommands
         OCTET STRING,
      xfsIDCCapabilitiesExtraCapability
         OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.2.1.7.1.1
xfsIDCCapabilitiesManagedServiceName OBJECT-TYPE
   SYNTAX DisplayString
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Instance identifier of the managed service."
   ::= { xfsIDCCapabilitiesEntry 1 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.2
xfsIDCCapabilitiesDeviceType OBJECT-TYPE
   SYNTAX IxfsIDCCapabilitiesDeviceType
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Identifies the type of device as following numeric values."
   ::= { xfsIDCCapabilitiesEntry 2 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.3
xfsIDCCapabilitiesCompoundDevice OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
          "It specifies whether the logical device is part of a compound physical
device. It is a TruthValue type field. Allowed values are."
        ::= { xfsIDCCapabilitiesEntry 3 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.4
     xfsIDCCapabilitiesReadTracks OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "Specifies the tracks that can be read by the ID card unit as a
combination of the following flags in numeric value."
        ::= { xfsIDCCapabilitiesEntry 4 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.5
     xfsIDCCapabilitiesWriteTracks OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "Specifies the tracks that can be written by the ID card unit as a
combination of the following flags in numeric value."
        ::= { xfsIDCCapabilitiesEntry 5 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.6
     xfsIDCCapabilitiesChipProtocols OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "Specifies the chip card protocols that are supported by the Service
Provider as a combination of the following flags as numeric variable."
        ::= { xfsIDCCapabilitiesEntry 6 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.7
     xfsIDCCapabilitiesRetCards OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "Specifies the maximum number of cards that can be held in the retain bin
as numeric variable."
        ::= { xfsIDCCapabilitiesEntry 7 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.8
     xfsIDCCapabilitiesSecurityType OBJECT-TYPE
        SYNTAX IxfsIDCCapabilitiesSecurityType
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "Specifies the type of security module used as one of the following
values as numeric variable."
        ::= { xfsIDCCapabilitiesEntry 8 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.9
     xfsIDCCapabilitiesPowerOnOption OBJECT-TYPE
        SYNTAX IxfsIDCCapabilitiesPowerOptions
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
          "Specifies the power-on capabilities of the device hardware as one of the
following values
           (applicable only to motor driven ID card units) as numeric variable."
        ::= { xfsIDCCapabilitiesEntry 9 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.10
     xfsIDCCapabilitiesPowerOffOption OBJECT-TYPE
        SYNTAX IxfsIDCCapabilitiesPowerOptions
```

```
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "Specifies the power-off capabilities of the device hardware as one of
the following values
           (applicable only to motor driven ID card units) as numeric variable."
        ::= { xfsIDCCapabilitiesEntry 10 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.11
     xfsIDCCapabilitiesProgrammableFlux OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "It specifies whether the flux sensor is programmable or not in
TruthValue format."
        ::= { xfsIDCCapabilitiesEntry 11 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.12
     xfsIDCCapabilitiesReadWriteAfterEject OBJECT-TYPE
        SYNTAX TruthValue
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "It specifies whether the card can be read or written to after ejected to
the exit slot in TruthValue format."
        ::= { xfsIDCCapabilitiesEntry 12 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.13
     xfsIDCCapabilitiesWriteMode OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "A combination of the following flags specify the write capabilities,
with respect to whether the
           device can write low coercivity (loco) and/or high coercivity (hico)
magnetic stripes. This is a numeric value."
        ::= { xfsIDCCapabilitiesEntry 13 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.14
     xfsIDCCapabilitiesChipPower OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "Specifies the capabilities of the ID card unit (in relation to the user
or permanent chip controlled
           by the service), for chip power management as a combination of the
following flags as numeric variable."
        ::= { xfsIDCCapabilitiesEntry 14 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.15
     xfsIDCCapabilitiesDipMode OBJECT-TYPE
        SYNTAX IxfsIDCCapabilitiesDIPMode
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
           "Specifies the capabilities of the ID card unit (in relation
           to the user or permanent chip controlled by the service), for
           chip power management."
        ::= { xfsIDCCapabilitiesEntry 15 }


     -- 1.3.6.1.4.1.16213.2.2.1.7.1.16
     xfsIDCCapabilitiesMemoryChipProtocols OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
```

```
      "Specifies the memory card protocols that are supported by
      the Service Provider."
   ::= { xfsIDCCapabilitiesEntry 16 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.17
xfsIDCCapabilitiesGuidanceCardUnit OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Specifies the guidance light capability of the card unit."
   ::= { xfsIDCCapabilitiesEntry 17 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.18
xfsIDCCapabilitiesEjectPosition OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Specifies the target position that is supported for the
      eject operation."
   ::= { xfsIDCCapabilitiesEntry 18 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.19
xfsIDCCapabilitiesPowerSaveControl OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "This TruthValue variable specifies whether or not power
      saving control is available."
   ::= { xfsIDCCapabilitiesEntry 19 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.20
xfsIDCCapabilitiesParkingStations OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Specifies the number of supported parking stations.
      If a zero value is specified there is no parking station
      supported."
   ::= { xfsIDCCapabilitiesEntry 20 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.21
xfsIDCCapabilitiesAntiFraudModule OBJECT-TYPE
   SYNTAX TruthValue
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Specifies whether the anti-fraud module is available."
   ::= { xfsIDCCapabilitiesEntry 21 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.22
xfsIDCCapabilitiesSynchronizableCommands OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
      "Specifies which Command IDs can be synchronized."
   ::= { xfsIDCCapabilitiesEntry 22 }


-- 1.3.6.1.4.1.16213.2.2.1.7.1.100
xfsIDCCapabilitiesExtraCapability OBJECT-TYPE
   SYNTAX OCTET STRING
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
```

```
          "Vendor dependent additional device capablities information."
        ::= { xfsIDCCapabilitiesEntry 100 }


     -- 1.3.6.1.4.1.16213.3.0
     xfsTrapV2 OBJECT-IDENTITY
        STATUS current
        DESCRIPTION
          "Root node for the converted TRAP-TYPEs."
        ::= { xfsTrap 0 }


-- ******************************************************************
-- Trap definitions
-- ******************************************************************
     -- 1.3.6.1.4.1.16213.3.0.102
     xfsIDCDetailedDSCTrap NOTIFICATION-TYPE
        OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
          xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapEvent,
          xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsIDCStatusDevice,
xfsIDCStatusNumberSubDevices, xfsIDCStatusMedia,
          xfsIDCStatusRetainBin, xfsIDCStatusSecurity, xfsIDCStatusCardRetained,
xfsIDCStatusChipPower, xfsIDCStatusExtraStatus,
          xfsIDCStatusGuidanceCardUnit, xfsIDCStatusChipModule,
xfsIDCStatusMagReadModule, xfsIDCStatusMagWriteModule,
xfsIDCStatusFrontImageModule,
          xfsIDCStatusBackImageModule, xfsIDCStatusDevicePosition,
xfsIDCStatusPowerSaveRecoveryTime, xfsIDCStatusParkingStationMedia,
xfsIDCStatusAntiFraudModule }
        STATUS current
        DESCRIPTION
          "This trap indicates a change in the status of a managed
          service."
        ::= { xfsTrapV2 102 }


     -- 1.3.6.1.4.1.16213.3.0.302
     xfsIDCResetDeviceCompleteTrap NOTIFICATION-TYPE
        OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
          xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapDate,
          xfsCommonTrapSPVersion, xfsIDCStatusDevice, xfsIDCStatusNumberSubDevices,
xfsIDCStatusMedia, xfsIDCStatusRetainBin,
          xfsIDCStatusSecurity, xfsIDCStatusCardRetained, xfsIDCStatusChipPower,
xfsIDCStatusExtraStatus, xfsIDCStatusGuidanceCardUnit,
          xfsIDCStatusChipModule, xfsIDCStatusMagReadModule,
xfsIDCStatusMagWriteModule, xfsIDCStatusFrontImageModule,
xfsIDCStatusBackImageModule,
          xfsIDCStatusDevicePosition, xfsIDCStatusPowerSaveRecoveryTime,
xfsIDCStatusParkingStationMedia, xfsIDCStatusAntiFraudModule }
        STATUS current
        DESCRIPTION
          "This trap indicates the Reset action has complete and reports the
          state of the device after the reset."
        ::= { xfsTrapV2 302 }


   END

--
-- SMIv2_xfsIDC.mib
--
```

5. Appendix B - C-Header files

5.1 XFSMIBIDC.H

```
/**********************************************************************
*                                                                     *
* xfsmibidc.h      CEN/XFS - MIB IDC                                  *
*                                                                     *
*               Version 3.30  --  Feb 6, 2017                         *
*                                                                     *
**********************************************************************/

#ifndef __inc_xfsmibidc__h
#define __inc_xfsmibidc__h

#ifdef __cplusplus
extern "C" {
#endif


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

} xfsIDCMediaStatus;


enum IxfsIDCRetainBinStatus
{
        xfsIDCRetainBinOK               = 2,
        xfsIDCRetainBinNotSupported,
        xfsIDCRetainBinFull,
        xfsIDCRetainBinHigh,
        xfsIDCRetainBinMissing

} xfsIDCRetainBinStatus;


enum IxfsIDCSecurityStatus
{
        xfsIDCSecurityNotSupported      = 2,
        xfsIDCSecurityNotReady,
        xfsIDCSecurityOpen

} xfsIDCSecurityStatus;


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

enum IxfsIDCChipModuleStatus
{
        xfsIDCChipModuleOK              = 2,
        xfsIDCChipModuleInop,
        xfsIDCChipModuleUnknown,
        xfsIDCChipModuleNotSupported
```

```
} xfsIDCChipModuleStatus;

enum IxfsIDCMagModuleStatus
{
        xfsIDCMagModuleOK               = 2,
        xfsIDCMagModuleInop,
        xfsIDCMagModuleUnknown,
        xfsIDCMagModuleNotSupported

} xfsIDCMagModuleStatus;

enum IxfsIDCImageModuleStatus
{
        xfsIDCImageModuleOK             = 2,
        xfsIDCImageModuleInop,
        xfsIDCImageModuleUnknown,
        xfsIDCImageModuleNotSupported

} xfsIDCImageModuleStatus;

enum IxfsIDCDevicePositionStatus
{
        xfsIDCDeviceInPosition          = 1,
        xfsIDCDeviceNotInPosition,
        xfsIDCDevicePosUnknown,
        xfsIDCDevicePosNotSupported
} xfsIDCDevicePositionStatus;

enum IxfsIDCAntiFraudModuleStatus
{
        xfsIDCAFMNotSupported           = 1,
        xfsIDCAFMOK,
        xfsIDCAFMInop,
        xfsIDCAFMDeviceDetected,
        xfsIDCAFMUnknown
} xfsIDCAntiFraudModuleStatus;


enum IxfsIDCCapabilitiesDeviceType
{
        xfsIDCTypeMotor                 = 2,
        xfsIDCTypeSwipe,
        xfsIDCTypeDip,
        xfsIDCTypeContactless,
        xfsIDCTypeLatchedDip,
        xfsIDCTypePermanent,
        xfsIDCTypeIntelligentContactless
} xfsIDCCapabilitiesDeviceType;

enum IxfsIDCCapabilitiesSecurityType
{
        xfsIDCSecNotSupported           = 2,
        xfsIDCSecMMBox,
        xfsIDCSecCIM86
} xfsIDCCapabilitiesSecurityType;

enum IxfsIDCCapabilitiesPowerOptions
{
        xfsIDCNoAction                  = 2,
        xfsIDCEject,
        xfsIDCRetain,
        xfsIDCEjectThenRetain,
        xfsIDCReadPosition
} xfsIDCCapabilitiesPowerOptions;

enum IxfsIDCCapabilitiesDIPMode
{
        xfsIDCNotSupported              = 1,
        xfsIDCDipUnknown                = 2,
        xfsIDCDipExit                   = 3,
        xfsIDCDipEntry                  = 5,
        xfsIDCDipEntryExit              = 9
} xfsIDCCapabilitiesDIPMode;
```

```
/*********************************************************************
*                                                                    *
*        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        xfsIDCStatusGuidanceCardUnit        (9)
#define        xfsIDCStatusChipModule              (10)
#define        xfsIDCStatusMagReadModule           (11)
#define        xfsIDCStatusMagWriteModule          (12)
#define        xfsIDCStatusFrontImageModule        (13)
#define        xfsIDCStatusBackImageModule         (14)
#define        xfsIDCStatusDevicePosition          (15)
#define        xfsIDCStatusPowerSaveRecoveryTime   (16)
#define        xfsIDCStatusParkingStationMedia     (17)
#define        xfsIDCStatusAntiFraudModule         (18)
#define        xfsIDCStatusExtraStatus             (100)


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

/*********************************************************************
*                                                                    *
*        MIB Variables for the Capabilities Table                    *
*                                                                    *
*********************************************************************/
#define        xfsIDCCapabilitiesManagedServiceName     (1)
#define        xfsIDCCapabilitiesDeviceType             (2)
#define        xfsIDCCapabilitiesCompoundDevice         (3)
#define        xfsIDCCapabilitiesReadTracks             (4)
#define        xfsIDCCapabilitiesWriteTracks            (5)
#define        xfsIDCCapabilitiesChipProtocols          (6)
#define        xfsIDCCapabilitiesRetCards               (7)
#define        xfsIDCCapabilitiesSecurityType           (8)
#define        xfsIDCCapabilitiesPowerOnOption          (9)
#define        xfsIDCCapabilitiesPowerOffOption         (10)
#define        xfsIDCCapabilitiesProgrammableFlux       (11)
#define        xfsIDCCapabilitiesReadWriteAfterEject    (12)
#define        xfsIDCCapabilitiesWriteMode              (13)
#define        xfsIDCCapabilitiesChipPower              (14)
#define        xfsIDCCapabilitiesDipMode                (15)
#define        xfsIDCCapabilitiesMemoryChipProtocols    (16)
#define        xfsIDCCapabilitiesGuidanceCardUnit       (17)
#define        xfsIDCCapabilitiesEjectPosition          (18)
#define        xfsIDCCapabilitiesPowerSaveControl       (19)
#define        xfsIDCCapabilitiesParkingStations        (20)
#define        xfsIDCCapabilitiesAntiFraudModule        (21)
#define        xfsIDCCapabilitiesSynchronizableCommands (22)
#define        xfsIDCCapabilitiesExtraCapability        (100)


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

#endif  /* __inc_xfsmibidc__h */
```