

**CEN**

**CWA 15748-45**

**WORKSHOP**

September 2011

**AGREEMENT**

---

ICS 35.240.40

English version

**Extensions for Financial Services (XFS) interface specification -  
Release 3.10 - Part 45: XFS MIB Device Specific Definitions -  
Card Dispenser Device Class MIB 3.10**

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

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

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

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

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



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

**Management Centre: Avenue Marnix 17, B-1000 Brussels**

---

© 2011 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.:CWA 15748-45:2011 D/E/F

## Table of Contents

---

<b>FOREWORD</b> .....	<b>3</b>
<b>1. INTRODUCTION</b> .....	<b>6</b>
<b>2. XFS CRD MIB VARIABLES</b> .....	<b>9</b>
2.1 XFS CRD Status Table .....	9
2.1.1 xfsCRDStatusTable: States .....	9
2.2 XFS CRD Sub Device Table.....	12
2.2.1 xfsCRDSubDeviceTable: .....	12
2.3 XFS CRD Error Table.....	13
2.4 XFS CRD Reset Table .....	14
2.5 XFS CRD Reset Device Table .....	14
2.6 XFS CRD Capabilities Table .....	15
2.6.1 xfsCRDCapabilitiesTable: Capabilities .....	15
<b>3. CRD TRAPS</b> .....	<b>18</b>
3.1 CRD Detailed Device Status Change Trap .....	18
3.1.1 CRD Detailed Device Status Change Trap Format .....	18
3.1.2 CRD Detailed Device Status Change Trap: an example .....	20
3.2 CRD Sub-Device Status Change Trap .....	21
3.2.1 CRD Sub-Device Status Change Trap Format .....	21
3.2.2 CRD Sub-Device Status Change Trap: an example .....	23
3.3 CRD Reset Device Complete Trap .....	25
3.3.1 CRD Reset Device Complete Trap Format .....	25
3.3.2 CRD Reset Device Complete: an example.....	27
<b>4. APPENDIX A - CRD MIB SUB-TREE</b> .....	<b>29</b>
4.1 CRD MIB in SMIv2 and SMIv1 ASN-1 format .....	29
<b>5. APPENDIX B - C-HEADER FILES</b> .....	<b>43</b>
5.1 XFSMIBCRD.H.....	43

## Foreword

---

This CWA is revision 3.10 of the XFS interface specification.

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties on 2007-11-29, 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 3.10.

A list of the individuals and organizations which supported the technical consensus represented by the CEN Workshop Agreement is available to purchasers from the CEN-CENELEC Management Centre. These organizations were drawn from the banking sector. The CEN/ISSS 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 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 MIB Version 3.10

Part 30: XFS MIB Device Specific Definitions - Printer Device Class MIB 3.10

Part 31: XFS MIB Device Specific Definitions - Identification Card Device Class MIB 3.10

Part 32: XFS MIB Device Specific Definitions - Cash Dispenser Device Class MIB 3.10

Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class MIB 3.10

Part 34: XFS MIB Device Specific Definitions - Check Reader/Scanner Device Class MIB 3.10

Part 35: XFS MIB Device Specific Definitions - Depository Device Class MIB 3.10

Part 36: XFS MIB Device Specific Definitions - Text Terminal Unit Device Class MIB 3.10

Part 37: XFS MIB Device Specific Definitions - Sensors and Indicators Unit Device Class MIB 3.10

Part 38: XFS MIB Device Specific Definitions - Camera Device Class MIB 3.10

## **CWA 15748-45:2011 (E)**

Part 39: XFS MIB Device Specific Definitions - Alarm Device Class MIB 3.10

Part 40: XFS MIB Device Specific Definitions - Card Embossing Unit Device Class MIB 3.10

Part 41: XFS MIB Device Specific Definitions - Cash-In Module Device Class MIB 3.10

Part 42: Reserved for future use.

Part 43: XFS MIB Device Specific Definitions - Vendor Dependent Mode Class MIB 3.10

Part 44: XFS MIB Application Management MIB 3.10

Part 45: XFS MIB Device Specific Definitions - Card Dispenser Device Class MIB 3.10

Part 46: XFS MIB Device Specific Definitions - Barcode Reader Device Class MIB 3.10

Part 47: XFS MIB Device Specific Definitions - Item Processing Module Device Class MIB 3.10

Parts 48 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 62: Printer and Scanning Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.02 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.03 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.01 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.02 (see CWA 14050) to Version 3.10 (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/cen/pages/default.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/ISSS 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 2010-06-17 and was successfully closed on 2010-12-22. The final text of this CWA was submitted to CEN for publication on 2011-01-27.

This CEN Workshop Agreement is publicly available as a reference document from the National Members of CEN: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and 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:

3.10	December 14, 2010	Initial release.
------	-------------------	------------------

## 1. Introduction

---

This document provides the device specific MIB definition (Management Information Base) variables for the xfsCRD 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 xfsCRD version one sub-tree is identified by:

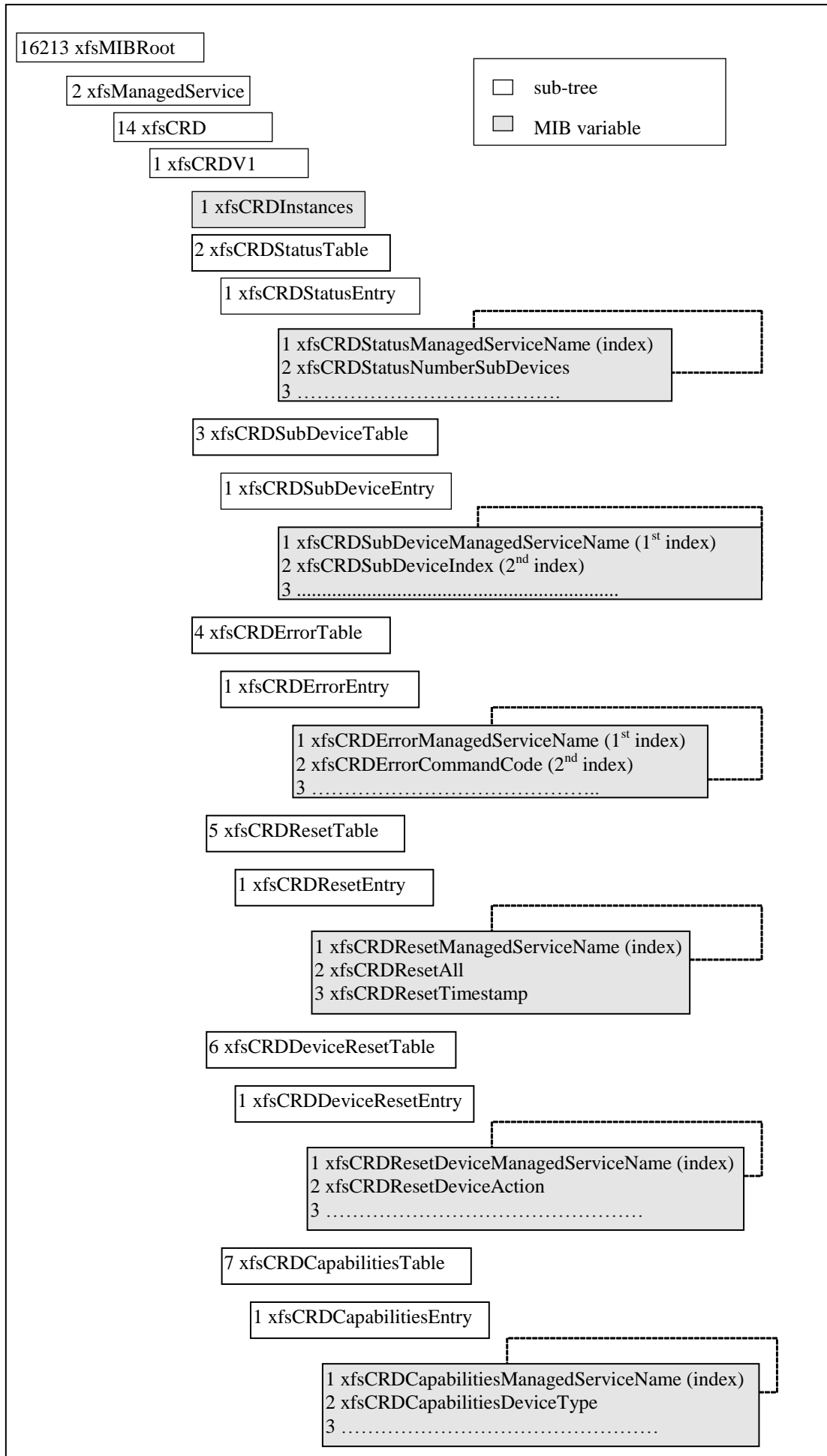
xfsMIBRoot

- xfsManagedService (2)
  - xfsCRD (14)
    - xfsCRDV1 (1)

The xfsCRDV1 sub-tree contains the following variables:

- *xfsCRDInstances(1)* is the number of managed services for the CRD class installed on the XFS subsystem. It is a 32 bit numerical field.
- *xfsCRDStatusTable(2)* identifies the table for the CRD variables.
- *xfsCRDSubDeviceTable(3)* this table contains the sub-device table for the CRD device.
- *xfsCRDErrorTable(4)* identifies the table for the CRD error counter variables.
- *xfsCRDResetTable(5)* identifies the table for the CRD reset variable.
- *xfsCRDResetDeviceTable(6)* identifies the table for the CRD reset device variables.
- *xfsCRDCapabilitiesTable(7)* identifies the table for the CRD 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 *xfsCRDV1* sub-tree:



## **CWA 15748-45:2011 (E)**

The next section describes how the Status, Sub-Device, Error, Reset, Reset Device and Capabilities tables apply to the CRD device class.



## 2. XFS CRD MIB variables

---

This section describes the MIB variables for the tables of the CRD Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Card Dispenser 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\_CRD\_RESET command to be executed from the management station.

### 2.1 XFS CRD Status Table

---

The *xfscRDStatusTable(2)* groups the variables identifying device status information, statistics and auxiliary variables. It is indexed through a single parameter, *xfscRDStatusManagedServiceName*. 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.

*xfscRDManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the CRD class.

As an example, the identifier for the device status value of *xfscRDStatusDispenser(4)* for a device with managed service name equal to "CardDispenser1" is as follows:

Character	C	a	r	d	D	i	s	p	e	n	s	e	r	1
ASCII Hex	43	61	72	64	44	69	73	70	65	6E	73	65	72	31
ASCII Dec	67	97	114	100	68	105	115	112	101	110	115	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:

*xfscMIBRoot.2.14.1.2.1.4.14.67.97.114.100.68.105.115.112.101.110.115.101114.49*

#### 2.1.1 xfscRDStatusTable: States

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

*xfscRDStatusManagedServiceName* (1)  
Uniquely identifies the managed service

*xfscRDStatusNumberSubDevices* (2)  
Defines how many sub-devices the service has.

*xfscRDStatusDevice* (3)  
It contains the state of the card dispenser device. Allowed values are:

Value	Meaning
<i>xfscDevOnline</i> (1)	The device is online. This is returned when the card dispenser is present and operational.
<i>xfscDevOffline</i> (2)	The device is offline (e.g. the operator has taken the device offline by turning a switch or pulling out the device).
<i>xfscDevPowerOff</i> (3)	The device is powered off or physically not connected.

## CWA 15748-45:2011 (E)

xfDevNoDevice(4)	The device is not intended to be there, e.g. this type of self-service machine does not contain such a device or it is internally not configured.
xfDevHWError(5)	The device is inoperable due to a hardware error.
xfDevUserError(6)	The device is present but a person is preventing proper device operation.
xfDevBusy(7)	The device is busy and unable to process an execute command at this time.
xfDevFraudAttempt(8)	The device is present but has detected a fraud attempt.

### xfCRDStatusDispenser (4)

It contains the state of the card units including all retain bins. Allowed values are:

Value	Meaning
xfCRDDispCUOK(1)	All card units present are in a good state.
xfCRDDispCUState(2)	The dispenser is operational, but one or more of the card units is in a low, empty or inoperative condition. Items can still be dispensed from at least one of the card units.
xfCRDDispCUStop(3)	Due to a card unit failure dispensing is impossible. The dispenser is operational, but no items can be dispensed because all of the card units are in an empty or inoperative condition.
xfCRDDispCUUnknown(4)	Due to a hardware error or other condition, the state of the card units cannot be determined.

### xfCRDStatusTransport (5)

It contains the state of the transport mechanism. Allowed values are:

Value	Meaning
xfCRDTransportOK(1)	The transport is in a good state
xfCRDTransportInop (2)	The transport is inoperative due to a hardware failure or media jam.
xfCRDTransportUnknown (3)	Due to a hardware error or other condition, the state of the transport cannot be determined.
xfCRDTransportNotSupported (4)	The physical device has no transport or transport state reporting is not supported.

### xfCRDStatusMedia (6)

It contains the state of a card that may or may not be present in the device. Allowed values are:

Value	Meaning
xfCRDMediaPresent(2)	The Media is present in the device, but not in the exiting position and not jammed.
xfCRDMediaNotPresent(3)	Media is not present in the device and not at the exiting position.
xfCRDMediaJammed(4)	Media is jammed in the device.
xfCRDMediaNotSupported(5)	Capability to report media position is not supported by the device.
xfCRDMediaUnknown(6)	The media state cannot be determined with the device in its current state.
xfCRDMediaExiting(7)	Media is at the exit slot of the card dispenser unit.

xfscRDMediaRetained(8) The Media has been retained.

xfscRDStatusShutter (7)

It contains the state of the shutter. Allowed values are:

Value	Meaning
xfscRDShutterClosed(1)	The shutter is closed.
xfscRDShutterOpen(2)	The shutter is open.
xfscRDShutterJammed(3)	The shutter is jammed.
xfscRDShutterUnknown(4)	Due to a hardware error or other condition, the state of the shutter cannot be determined.
xfscRDShutterNotSupported(5)	The physical device has no shutter or shutter state reporting is not supported.

xfscRDStatusGuidanceCardDisp (8)

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

Value	XFS Name	Meaning
0x00000000	WFS_CRD_GUIDANCE_NOT_AVAILABLE	Guidance is not available.
0x00000001	WFS_CRD_GUIDANCE_OFF	Guidance is off.
0x00000004	WFS_CRD_GUIDANCE_SLOW_FLASH	Guidance state is slow flash.
0x00000008	WFS_CRD_GUIDANCE_MEDIUM_FLASH	Guidance state is medium flash.
0x00000010	WFS_CRD_GUIDANCE_QUICK_FLASH	Guidance state is quick flash.
0x00000080	WFS_CRD_GUIDANCE_CONTINUOUS	Guidance state is continuous.
0x00000100	WFS_CRD_GUIDANCE_RED	Guidance is red.
0x00000200	WFS_CRD_GUIDANCE_GREEN	Guidance is green.
0x00000400	WFS_CRD_GUIDANCE_YELLOW	Guidance is yellow.
0x00000800	WFS_CRD_GUIDANCE_BLUE	Guidance is blue.
0x00001000	WFS_CRD_GUIDANCE_CYAN	Guidance is cyan.
0x00002000	WFS_CRD_GUIDANCE_MAGENTA	Guidance is magenta.
0x00004000	WFS_CRD_GUIDANCE_WHITE	Guidance is white.

xfscRDStatusDevicePosition (9)

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

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

**xfscRDStatusPowerSaveRecoveryTime (10)**

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.

**xfscRDStatusExtraStatus (100)**

It contains vendor dependent additional device status information. It is 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 CRD Sub Device Table

---

The *xfscRDSubDeviceTable(3)* groups the variables identifying information for the card units. It is indexed through two values, *xfscRDSubDeviceManagedServiceName* and *xfscRDSubDeviceIndex*. All sub-device variables are read-only.

*xfscRDSubDeviceManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the CRD class. As an example, the identifier for the sub-device status value of *xfscRDSubDeviceCUCardName(3)* for sub-device index 1 on a device with managed service name equal to "CardDispenser1" is as follows:

Character	C	a	r	d	D	i	s	p	e	n	s	e	r	1
ASCII Hex	43	61	72	64	44	69	73	70	65	6E	73	65	72	31
ASCII Dec	67	97	114	100	68	105	115	112	101	110	115	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:

*xfscMIBRoot.2.14.132.1.4.14.67.97.114.100.68.105.115.112.101.110.115.101.114.49.1*

### 2.2.1 xfscRDSubDeviceTable:

The first two variables are common across all devices, the other variables are sub-device specific.

It should be noted that in XFS the CRD information for the CRD MIB Sub Device Table comes from the XFS CRD command *WFS\_INF\_CRD\_CARD\_UNIT\_INFO*.

**xfscRDSubDeviceManagedServiceName (1)**

Uniquely identifies the managed service.

**xfscRDSubDeviceIndex (2)**

Index to the sub-device table only. This variable has no relationship to the card unit. This is an index (starting from 1) into the CRD Sub-device table.

**xfscRDSubDeviceCUCardName (3)**

An identifier that is used to identify the type of cards in the card unit. This is an OCTET STRING.

**xfscRDSubDeviceCUType (4)**

Type of card unit as one of the following values:

Value	Meaning
<i>xfscRDTypeSupplyBin(2)</i>	The card unit is a supply card unit.
<i>xfscRDTypeRetainBin(3)</i>	The card unit is a retain card unit.

**xfscRDSubDeviceCUInitialCount (5)**

Initial number of items contained in the card unit.

**xfscRDSubDeviceCUCount (6)**

The number of items inside the card unit plus any items from the card units not yet presented to the customer. This count is decremented when the items are either presented to the customer or retained. This count is incremented for a retain bin after a retain operation.

**xfscRDSubDeviceCUREtainCount (7)**

The number of items from this card unit which are in the retain bin. This field is always zero for a retain bin. This value is persistent.

*xfsCRDSubDeviceCUThreshold* (8)

When *ulCount* reaches this value the XFS WFS\_USRE\_CRD\_CARDUNITTHRESHOLD threshold event will be generated. If this value is non-zero then hardware sensors in the device do not trigger threshold events.

*xfsCRDSubDeviceCUStatus* (9)

It contains the status of the card unit as one of the following values and is a numeric type field. Allowed values are:

Value	Meaning
<i>xfsCRDStatCUOK</i> (1)	The card supply or retain unit is in a good state.
<i>xfsCRDStatCULow</i> (2)	The card supply unit is almost empty.
<i>xfsCRDStatCUEmpty</i> (3)	The card supply unit is empty.
<i>xfsCRDStatCUInop</i> (4)	The card supply or retain unit is inoperative.
<i>xfsCRDStatCUMissing</i> (5)	The card supply unit is missing.
<i>xfsCRDStatCUHigh</i> (6)	The retain card unit is almost full.
<i>xfsCRDStatCUFull</i> (7)	The retain card unit is full.
<i>xfsCRDStatCUUnknown</i> (8)	The status of the card unit cannot be determined.

*xfsCRDSubDeviceCUHardwareSensors* (10)

This will either be FALSE or TRUE. It is a TruthValue (RFC1253-MIB) where 1 = TRUE and 2 = FALSE. Specifies whether or not threshold events can be generated based on hardware sensors in the device. This applies to XFS WFS\_CRD\_STATCULOW and WFS\_CRD\_STATCUHIGH thresholds only. If this value is TRUE then threshold events may be generated based on hardware sensors as opposed to counts. If the XFS *ulThreshold* value is non zero then hardware triggers are ignored and software trigger/counters are used. An XFS WFS\_CRD\_STATCUHIGH threshold will be sent for a retain bin or WFS\_CRD\_STATCULOW for a card supply unit.

## 2.3 XFS CRD Error Table

The *xfsCRDErrorTable*(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

*xsfCRDErrorManagedServiceName*  
*xfsCRDErrorCommandCode*  
*xfsCRDErrorResponseCode*

The *xfsCRDErrorTable* is defined as:

- *xfsCRDErrorManagedServiceName*(1) which provides the primary index to the service in question. It is Display String field. The *xfsCRDErrorManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "CardDispenser1".
- *xfsCRDErrorCommandCode*(2) is an index which identifies the command code that that response code is related to, e.g. WFS\_CMD\_CRD\_DISPENSE\_CARD (1401). It is a 32 bit numerical field.
- *xfsCRDErrorResponseCode*(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\_CRD\_NOMEDIA (-1411) is represented by 1411. It is a 32 bit numerical field.
- *xfsCRDErrorCount*(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\_CRD\_NOMEDIA (-1411) error returned from the WFS\_CMD\_CRD\_EJECT\_CARD (1402) command for a device with managed service name equal to “CardDispenser1” is as follows:

*xfsMIBRoot.2.14.1.4.1.4.14.67.97.114.100.68.105.115.112.101.110.115.101.114.49.1402.1411*

## 2.4 XFS CRD Reset Table

---

The *xfsCRDResetTable(5)* contains the *xfsCRDResetAll* and *xfsCRDResetTimestamp* variables and is indexed by the single variable, *xfsCRDResetManagedServiceName*. When the *xfsCRDResetAll* 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 *xfsCRDResetTable* is defined as:

- *xfsCRDResetManagedServiceName(1)* which provides the primary index to the service in question. It is Display String field. The *xfsCRDResetManagedServiceName* parameter corresponds to the value of *xfsCRDRoot.xfsGeneral.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. “CardDispenser1”.
- *xfsCRDResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsCRDResetAll* 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 *xfsCRDResetAll* variable will return 0 (zero).
- *xfsCRDResetTimestamp(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 “CardDispenser1” by setting the value zero in the *xfsCRDResetAll* variable represented by:

*xfsMIBRoot.2.14.1.5.1.2.14.67.97.114.100.68.105.115.112.101.110.115.101.114.49*

## 2.5 XFS CRD Reset Device Table

---

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

The *xfsCRDResetDeviceAction* 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\_CRD\_RESET command will be issued.
4. Exclusive access to the device will be relinquished when the WFS\_CMD\_CRD\_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 *xfsCRDResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request.

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

The *xfsCRDResetDeviceTable* is defined as:

- *xfsCRDResetDeviceManagedServiceName(1)* which provides the index to the service in question. It is a Display String field. The *xfsCRDResetDeviceManagedServiceName* parameter corresponds to the value of

*xfsmIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. “CardDispenser1”.

- *xfscRDResetDeviceAction(2)* is a read-write variable. Issue of a Set command on the *xfscRDResetDeviceAction* variable with value *executeReset(1)* will result in the device being reset as described above.
- *xfscRDResetDeviceMediaControl(3)* is a read only variable. This variable reports how any media found within the device is handled. The value of the *xfscRDResetDeviceMediaControl* 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. The detailed device specific media control information (e.g. CRD retract area to retract media to) is configured through local SNMP Agent configuration.
- *xfscRDResetDeviceStatus(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 “CardDispenser1” is reset by setting the *xfscRDResetDeviceAction* variable represented by:

*xfsmIBRoot.2.14.1.6.1.2.14.67.97.114.100.68.105.115.112.101.110.115.101.114.49*

## 2.6 XFS CRD Capabilities Table

The *xfscRDCapabilitiesTable(7)* groups the variables identifying device capabilities information and auxiliary variables. It is indexed through a single parameter, *xfscRDCapabilitiesManagedServiceName*. 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.

*xfscRDCapabilitiesManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the CRD class.

As an example, the identifier for the device capabilities value of *xfscRDCapabilitiesCompound(2)* for a device with managed service name equal to “CardDispenser1” is as follows:

Character	C	a	r	d	D	i	s	p	e	n	s	e	r	1
ASCII Hex	43	61	72	64	44	69	73	70	65	6E	73	65	72	31
ASCII Dec	67	97	114	100	68	105	115	112	101	110	115	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.2.14.1.7.1.2.13.67.97.115.104.65.99.99.101.112.116.111.114.4*

### 2.6.1 xfscRDCapabilitiesTable: Capabilities

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

*xfscRDCapabilitiesManagedServiceName (1)*  
Uniquely identifies the managed service.

*xfscRDCapabilitiesCompound (2)*  
Specifies if the logical device is part of a compound device in a TruthValue variable as follows:

Value	Meaning
True(1)	The device is a compound device.
False(2)	The device is not a compound device.

**xfscRDCapabilitiesPowerOnOption (3)**

Specifies the power-on capabilities of the device hardware, as one of the following flags; applicable only to motor driven ID card units:

Value	Meaning
xfscRDPwrNoAction(2)	No power on actions are supported by the device.
xfscRDPwrEject(3)	The card will be ejected on power-on (or off).
xfscRDPwrRetain(4)	The card will be retained on power-on (off).
xfscRDPwrEjectThenRetain(5)	The card will be ejected for a specified time after power-on then retained if not taken. The time for which the card is ejected is vendor dependent.

**xfscRDCapabilitiesPowerOffOption (4)**

Specifies the power-off capabilities of the device hardware, as one of the one of the flags specified for *xfscRDCapabilitiesPowerOnOption*:

**xfscRDCapabilitiesCardTakenSensor (5)**

Specifies whether or not the card dispenser has the ability to detect when a card is taken from the exit slot by a user in a TruthValue variable as follows:

Value	Meaning
True(1)	The device has a taken sensor.
False(2)	The device does not have a taken sensor.

**xfscRDCapabilitiesDispenseTo (6)**

Specifies where a card will be dispensed to. Possible values are reported as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x00000000	WFS_CRD_DISPTO_CONSUMER	A dispensed card can be delivered to the exit slot for the consumer to take.
0x00000001	WFS_CRD_DISPTO_TRANSPORT	A dispensed card can be delivered into the transport mechanism.



## xfsCRDCapabilitiesGuidanceCardDisp (7)

It contains the capability of the guidance light. Possible states are reported as a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x00000000	WFS_CRD_GUIDANCE_NOT_AVAILABLE	There is no guidance control available at this position.
0x00000001	WFS_CRD_GUIDANCE_OFF	The light can be off.
0x00000004	WFS_CRD_GUIDANCE_SLOW_FLASH	The light can blink slowly.
0x00000008	WFS_CRD_GUIDANCE_MEDIUM_FLASH	The light can blink medium frequency.
0x00000010	WFS_CRD_GUIDANCE_QUICK_FLASH	The light can blink quickly.
0x00000080	WFS_CRD_GUIDANCE_CONTINUOUS	The light can be continuous (steady).
0x00000100	WFS_CRD_GUIDANCE_RED	The light can be red.
0x00000200	WFS_CRD_GUIDANCE_GREEN	The light can be green.
0x00000400	WFS_CRD_GUIDANCE_YELLOW	The light can be yellow.
0x00000800	WFS_CRD_GUIDANCE_BLUE	The light can be blue.
0x00001000	WFS_CRD_GUIDANCE_CYAN	The light can be cyan.
0x00002000	WFS_CRD_GUIDANCE_MAGENTA	The light can be magenta.
0x00004000	WFS_CRD_GUIDANCE_WHITE	The light can be white.

## xfsCRDCapabilitiesPowerSaveControl (8)

It contains 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.

## xfsCRDCapabilitiesExtraCapability (100)

It contains 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. CRD Traps

---

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

#### 3.1 CRD 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 CRD 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 CRD reflect the CRD 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\_CRD\_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 113 defines the trap as a CRD 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 CRD Detailed Device Status Change Trap Format

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

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

This variable binding contains the system generating the alarm; it is a Display String field. It corresponds to *lpszWorkstationName* in the device status change event data from the Service Provider.

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

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

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

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

HKEY\_LOCAL\_MACHINE\SOFTWARE\XFS\MANAGEMENT\_PROVIDERS\<ManagedServiceName>\class. This ID matches the class OID branch number i.e. CIM=1, IDC=2, CIM=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\_CRD\_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 CRD MIB class is represented by .1.3.6.1.4.1.16213.2.13

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.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusNumberSubDevices.xfsCRDStatusManagedServiceName (13)**

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

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusDevice.xfsCRDStatusManagedServiceName (14)**

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

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusDispenser.xfsCRDStatusManagedServiceName (15)**

It contains the state of the card units including all retain bins. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusTransport.xfsCRDStatusManagedServiceName (16)**

It contains the state of the transport mechanism. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusMedia.xfsCRDStatusManagedServiceName (17)**

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

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusShutter.xfsCRDStatusManagedServiceName (18)**

It contains the state of the shutter. It is a numeric type field.

## CWA 15748-45:2011 (E)

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.**xfCRDStatusGuidanceCardDisp**.xfsCRDStatusManagedServiceName (19)

It contains the state of the Guidance Light. It is an INTEGER field.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.**xfCRDDevicePosition**.xfsCRDStatusManagedServiceName (20)

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

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.**xfCRDStatusPowerSaveRecoveryTime**.xfsCRDStatusManagedServiceName (21)

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.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.**xfCRDStatusExtraStatus**.xfsCRDStatusManagedServiceName (22)

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

### 3.1.2 CRD Detailed Device Status Change Trap: an example

As an example, the following variable binding list represents a detailed device status change trap (6, 113) that is generated for a CRD with a managed service name of "CardDispenser1". It reports that the device is OFFLINE because the Dispenser is stopped.

xfMIBRoot.3.1.3.1	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName) "SST System 1"
xfMIBRoot.3.1.3.2	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName) "CardDispenser1"
xfMIBRoot.3.1.3.3	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass) 13 (WFS_SERVICE_CLASS_CRD)
xfMIBRoot.3.1.3.4	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName) "CRD"
xfMIBRoot.3.1.3.5	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType) 1 (WFS_CRD)
xfMIBRoot.3.1.3.6	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid) " 1.3.6.1.4.1.16213.2.14"
xfMIBRoot.3.1.3.7	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName) "ABC Corp Card Dispenser"
xfMIBRoot.3.1.3.8	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor) "Best Device Incorporated"
xfMIBRoot.3.1.3.9	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion) "1.10"
xfMIBRoot.3.1.3.10	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent) 4 (WFS_SYSE_DEVICE_STATUS)
xfMIBRoot.3.1.3.11	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate) "20/03/2003 15:40:53 -300"

xfMIBRoot.3.1.3.12	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapSPVersion)
	"1.23"
xfMIBRoot.2.14.1.2.1.3.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusDevice.xfCRDStatusManagedServiceName)
	2 (WFS_STAT_DEVOFFLINE)
xfMIBRoot.2.14.1.2.1.2.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusNumberSubDevices.xfCRDStatusManagedServiceName)
	1 (One sub device)
xfMIBRoot.2.14.1.2.1.4.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusDispenser.xfCRDStatusManagedServiceName)
	1 (xfCRDDispCUOK)
xfMIBRoot.2.14.1.2.1.5.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusTransport.xfCRDStatusManagedServiceName)
	1 (xfCRDDispCUOK)
xfMIBRoot.2.14.1.2.1.6.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusMedia.xfCRDStatusManagedServiceName)
	2 (xfCRDMediaNotPresent)
xfMIBRoot.2.14.1.2.1.7.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusShutter.xfCRDStatusManagedServiceName)
	1 (xfCRDShutterClosed)
xfMIBRoot.2.14.1.2.1.8.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusGuidanceCardDisp.xfCRDStatusManagedServiceName)
	1 (value corresponding to WFS_CRD_GUIDANCE_OFF)
xfMIBRoot.2.14.1.2.1.9.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusDevicePosition.xfCRDStatusManagedServiceName)
	1 (xfCRDDeviceInPosition)
xfMIBRoot.2.14.1.2.1.10.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusPowerSaveRecoveryTime.xfCRDStatusManagedServiceName)
	3 (3 seconds to recover from power saving mode)
xfMIBRoot.2.14.1.2.1.100.Index	(xfMIBRoot.xfManagedService.xfCRD.xfCRDV1.xfCRDStatusTable.xfCRDStatusEntry.xfCRDStatusExtraStatus.xfCRDStatusManagedServiceName)
	"\0"\0" ( No extra data )

## 3.2 CRD Sub-Device Status Change Trap

On the CRD device class the Sub Device Status change traps are sent when a WFS\_SRVE\_CRD\_CARDUNITINFOCHANGED event is generated. This trap is sent in addition to the threshold event defined in the architecture specification.

The definition of the content of the device specific fields within the Sub-Device Status trap is defined in section [2.2](#).

The SNMP Specific trap value 214 defines the trap as a CRD Sub-Device Status Change trap.

### 3.2.1 CRD Sub-Device Status Change Trap Format

The following defines the variable bindings included in the CRD Sub-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.

xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapManagedServiceName (1)

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` (2)

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

`HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<ManagedServiceName>\class`. This ID matches the class OID branch number i.e. CIM=1, IDC=2, CIM=3, etc.

`xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName` (3)

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` (4)

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_CRD_CAPABILITIES.fwType` field.

`xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid` (5)

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 CRD MIB class is represented by `.1.3.6.1.4.1.16213.2.14`

`xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName` (6)

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` (7)

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` (8)

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` (9)

The XFS event generating the alarm, it is a 32-bit integer (INT32). It corresponds to the message identifier associated with the XFS event generated by the Service Provider. For the CRD this corresponds to the `WFS_USRE_CRD_CARDUNITTHRESHOLD` event.

`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.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceIndex**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (12)  
Index identifying the sub-device.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceCUCardName**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (13)  
An identifier that is used to identify the type of cards in the card unit. It is an OCTET\_STRING field.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDSu  
bDeviceCUType**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (14)  
Type of items the card in unit take. It is a numeric type field.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDSu  
bDeviceCUInitialCount**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (15)  
Initial number of cards contained in the card unit. It is an INTEGER value.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceCUCount**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (16)  
Actual count of cards contained in the card unit. It is an INTEGER value.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceCUREtainCount**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (17)  
The number of items from this card unit which are in the retain bin. It is an INTEGER value.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceCUThreshold**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (18)  
The currently configured threshold value for this card unit. It is an INTEGER value.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceCUSTatus**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (19)  
Supplies the status of the card unit. It is a numeric type field.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.**xfCRDS  
ubDeviceCUHardwareSensor**.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex (20)  
Specifies whether or not threshold events can be generated based on hardware sensors in the device. This field is a TruthValue.

### 3.2.2 CRD Sub-Device Status Change Trap: an example

As an example, the following variable binding list represents a CRD sub-device status change trap (6, 213) generated from a generic XFS SST system. This trap sends an alarm to the SNMP Manager when a WFS\_SRVE\_CRD\_CARDUNITINFOCHANGED event is generated.

xfMIBRoot.3.1.3.2	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	"CardDispenser1"
xfMIBRoot.3.1.3.3	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	13 (WFS_SERVICE_CLASS_CRD)
xfMIBRoot.3.1.3.4	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	"CRD"
xfMIBRoot.3.1.3.5	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	1 (WFS_CRD_CRD)

**CWA 15748-45:2011 (E)**

xfsMIBRoot.3.1.3.6	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	“1.3.6.1.4.1.16213.2.14”
xfsMIBRoot.3.1.3.7	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	“ABC Corp Card Dispenser”
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)
	104 (WFS_SRVE_CRD_CARDUNITINFOCHANGED)
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.2.14.1.3.1.2.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceIndex.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	1 (Index to first sub device)
xfsMIBRoot.2.14.1.3.1.3.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUCardName.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	“LoyaltyCards”
xfsMIBRoot.2.14.1.3.1.4.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUType.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	1 (xfsCRDTypeSupplyBin)
xfsMIBRoot.2.14.1.3.1.5.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUInitialCount.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	1000 (1000 cards)
xfsMIBRoot.2.14.1.3.1.6.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUCount.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	928 (928 cards)
xfsMIBRoot.2.14.1.3.1.7.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUREtainCount.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	23 (23 cards)
xfsMIBRoot.2.14.1.3.1.8.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUThreshold.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	10 ( 10 cards)
xfsMIBRoot.2.13.1.3.1.9.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUStatus.xfsCRDSubDeviceManagedServiceName.xfsCRDSubDeviceIndex)
	1 (xfsCRDStatCUOK)
xfsMIBRoot.2.13.1.3.1.10.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDSubDeviceTable.xfsCRDSubDeviceEntry.xfsCRDSubDeviceCUHardwareSensor.xfsCRDSubDeviceManagedServiceName)



	me.xfsCRDSubDeviceIndex)
	FALSE (no hardware sensors )

### 3.3 CRD Reset Device Complete Trap

On the CRD 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 313 defines the trap as a CRD Reset Device Complete trap.

#### 3.3.1 CRD Reset Device Complete Trap Format

The following defines the variable bindings included in the CRD 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 **xfsCRDStatusDevice** binding ( var bind 12 below).

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

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

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

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

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

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

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

HKEY\_LOCAL\_MACHINE\SOFTWARE\XFS\MANAGEMENT\_PROVIDERS\<ManagedServiceName>\class.

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

This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS\_INF\_CRD\_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. The card dispenser module MIB class is represented by .1.3.6.1.4.1.16213.2.14

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.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusDevice.xfsCRDStatusManagedServiceName (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.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusNumberSubDevices.xfsCRDStatusManagedServiceName (13)**

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

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusDispenser.xfsCRDStatusManagedServiceName (14)**

It contains the state of the card units including all retain bins. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusTransport.xfsCRDStatusManagedServiceName (15)**

It contains the state of the transport mechanism. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusMedia.xfsCRDStatusManagedServiceName (16)**

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

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusShutter.xfsCRDStatusManagedServiceName (17)**

It contains the state of the shutter. It is a numeric type field.

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusGuidanceCardDisp.xfsCRDStatusManagedServiceName (18)**

It contains the state of the Guidance Light. It is an INTEGER field.

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDDevicePosition.xfsCRDStatusManagedServiceName (19)**

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

**xfsmIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusPowerSaveRecoveryTime.xfsCRDStatusManagedServiceName (20)**

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.

xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusExtraStatus.xfsCRDStatusManagedServiceName (21)

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

### 3.3.2 CRD Reset Device Complete: an example

As an example, the following variable binding list represents a Reset Device Complete trap (6, 313) generated as a result of a request to reset the device from the remote management station. The device in question is of type self-service bill with a managed service name "CardDispenser1".

xfMIBRoot.3.1.3.14	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult)
	0 (resetExecuted)
xfMIBRoot.3.1.3.2	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	"CardDispenser1"
xfMIBRoot.3.1.3.3	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	13 (WFS_SERVICE_CLASS_CRD)
xfMIBRoot.3.1.3.4	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	"CRD"
xfMIBRoot.3.1.3.5	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	1 (WFS_CRD)
xfMIBRoot.3.1.3.6	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	"1.3.6.1.4.1.16213.2.14"
xfMIBRoot.3.1.3.7	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	"ABC Corp Card Dispenser"
xfMIBRoot.3.1.3.8	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	"Best Device Incorporated"
xfMIBRoot.3.1.3.9	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	"1.10"
xfMIBRoot.3.1.3.11	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	"20/03/2003 15:40:53 -300"
xfMIBRoot.3.1.3.12	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	"1.23"
xfMIBRoot.2.14.1.2.1.3.Index	(xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusDevice.xfsCRDStatusManagedServiceName)
	2 (WFS_STAT_DEVOFFLINE)
xfMIBRoot.2.14.1.2.1.2.Index	(xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfsCRDStatusNumberSubDevices.xfsCRDStatusManagedServiceName)
	1 (One sub device)
xfMIBRoot.2.14.1.2.	(xfMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntr

**CWA 15748-45:2011 (E)**

1.4.Index	y.xfs <b>CRDStatusDispenser</b> .xfsCRDStatusManagedServiceName)
	1 (xfsCRDDispCUOK)
xfsMIBRoot.2.14.1.2. 1.5.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusTransport</b> .xfsCRDStatusManagedServiceName)
	1 (xfsCRDDispCUOK)
xfsMIBRoot.2.14.1.2. 1.6.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusMedia</b> .xfsCRDStatusManagedServiceName)
	2 (xfsCRDMediaNotPresent)
xfsMIBRoot.2.14.1.2. 1.7.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusShutter</b> .xfsCRDStatusManagedServiceName)
	1 (xfsCRDShutterClosed)
xfsMIBRoot.2.14.1.2. 1.8.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusGuidanceCardDisp</b> .xfsCRDStatusManagedServiceName)
	1 (value corresponding to WFS_CRD_GUIDANCE_OFF)
xfsMIBRoot.2.14.1.2. 1.9.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusDevicePosition</b> .xfsCRDStatusManagedServiceName)
	1 (xfsCRDDeviceInPosition)
xfsMIBRoot.2.14.1.2. 1.10.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusPowerSaveRecoveryTime</b> .xfsCRDStatusManagedServiceName)
	3 (3 seconds to recover from power saving mode)
xfsMIBRoot.2.14.1.2. 1.100.Index	(xfsMIBRoot.xfsManagedService.xfsCRD.xfsCRDV1.xfsCRDStatusTable.xfsCRDStatusEntry.xfs <b>CRDStatusExtraStatus</b> .xfsCRDStatusManagedServiceName)
	"\0"\0" ( No extra data )

## 4. Appendix A - CRD MIB sub-tree

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

### 4.1 CRD MIB in SMIv2 and SMIv1 ASN-1 format



SMIv1\_xfsCRD.mib



SMIv2\_xfsCRD.mib

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

```
-- *****
-- XFS MIB for CRD
-- Management Information Base for XFS CRD Device
--
-- The CRD Number is 14
-- The ASN.1 prefix to, and including the CRD is:
--     1.3.6.1.4.1.16213.2.14
-- *****

XFS-CRD-MIB DEFINITIONS ::= BEGIN

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

--
-- Type definitions
--

-- Type definitions
--
-- *****
-- CRD Status #defines
-- *****
    IxfsCRDDispenserStatus ::= INTEGER
    {
        xfsCRDDispCUOK(1),
        xfsCRDDispCUState(2),
        xfsCRDDispCUStop(3),
        xfsCRDDispCUUnknown(4)
    }

    IxfsCRDTransportStatus ::= INTEGER
    {
        xfsCRDTransportOK(1),
        xfsCRDTransportInop(2),
        xfsCRDTransportUnknown(3),
        xfsCRDTransportNotSupported(4)
    }

    IxfsCRDMediaStatus ::= INTEGER
    {
        xfsCRDMediaPresent(2),
        xfsCRDMediaNotPresent(3),
        xfsCRDMediaJammed(4),
        xfsCRDMediaNotSupported(5),
        xfsCRDMediaUnknown(6),
        xfsCRDMediaExiting(7),
        xfsCRDMediaRetained(8)
    }

```

## CWA 15748-45:2011 (E)

```
IxfsCRDShutterStatus ::= INTEGER
{
  xfsCRDShutterClosed(1),
  xfsCRDShutterOpen(2),
  xfsCRDShutterJammed(3),
  xfsCRDShutterUnknown(4),
  xfsCRDShutterNotSupported(5)
}

IxfsCRDDevicePositionStatus ::= INTEGER
{
  xfsCRDDeviceInPosition(1),
  xfsCRDDeviceNotInPosition(2),
  xfsCRDDevicePosUnknown(3),
  xfsCRDDevicePosNotSupported(4)
}

-- *****
-- CRD SubDevice #defines
-- *****
IxfsCRDCUType ::= INTEGER
{
  xfsCRDTypeSupplyBin(2),
  xfsCRDTypeRetainBin(3)
}

IxfsCRDCUStatus ::= INTEGER
{
  xfsCRDStatCUOK(1),
  xfsCRDStatCULow(2),
  xfsCRDStatCUEmpty(3),
  xfsCRDStatCUInop(4),
  xfsCRDStatCUMissing(5),
  xfsCRDStatCUHigh(6),
  xfsCRDStatCUFull(7),
  xfsCRDStatCUUnknown(8)
}

-- *****
-- CRD Capability #defines
-- *****
IxfsCRDPowerOptionCapabilities ::= INTEGER
{
  xfsCRDPwrNoAction(2),
  xfsCRDPwrEject(3),
  xfsCRDPwrRetain(4),
  xfsCRDPwrEjectThenRetain(5)
}

--
-- Node definitions
--

-- *****
-- Version 1 of CRD MIB
--
-- The ASN.1 prefix to, and including the Version 1 of CRD is:
1.3.6.1.4.1.16213.2.14.1
--
-- *****
-- 1.3.6.1.4.1.16213.2.14.1
xfsCRDV1 OBJECT IDENTIFIER ::= { xfsCRD 1 }

-- 1.3.6.1.4.1.16213.2.14.1.1
xfsCRDInstances OBJECT-TYPE
SYNTAX Integer32
```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Number that represents the number of CRD managed services."
 ::= { xfsCRDV1 1 }

-- *****
-- CRD Device Status Table
-- *****
-- 1.3.6.1.4.1.16213.2.14.1.2
xfsCRDStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsCRDStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the CRD status table."
    ::= { xfsCRDV1 2 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1
xfsCRDStatusEntry OBJECT-TYPE
    SYNTAX XfsCRDStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "CRD Device Status Table Entry."
    INDEX { xfsCRDStatusManagedServiceName }
    ::= { xfsCRDStatusTable 1 }

XfsCRDStatusEntry ::=
    SEQUENCE {
        xfsCRDStatusManagedServiceName
            DisplayString,
        xfsCRDStatusNumberSubDevices
            Integer32,
        xfsCRDStatusDevice
            IxfsMIBDeviceStatus,
        xfsCRDStatusDispenser
            IxfsCRDDispenserStatus,
        xfsCRDStatusTransport
            IxfsCRDTransportStatus,
        xfsCRDStatusMedia
            IxfsCRDMediaStatus,
        xfsCRDStatusShutter
            IxfsCRDShutterStatus,
        xfsCRDStatusGuidanceCardDisp
            Integer32,
        xfsCRDStatusDevicePosition
            IxfsCRDDevicePositionStatus,
        xfsCRDStatusPowerSaveRecoveryTime
            Integer32,
        xfsCRDStatusExtraStatus
            OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.1
xfsCRDStatusManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsCRDStatusEntry 1 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.2
xfsCRDStatusNumberSubDevices OBJECT-TYPE
    SYNTAX Integer32

```

## CWA 15748-45:2011 (E)

```
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Number of sub devices supported by the CRD device."
 ::= { xfsCRDStatusEntry 2 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.3
xfsCRDStatusDevice OBJECT-TYPE
    SYNTAX IxfsMIBDeviceStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Device status."
    ::= { xfsCRDStatusEntry 3 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.4
xfsCRDStatusDispenser OBJECT-TYPE
    SYNTAX IxfsCRDDispenserStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Dispenser Status.
         xfsCRDDispCUOK(1),
         xfsCRDDispCUState(2),
         xfsCRDDispCUStop(3),
         xfsCRDDispCUUnknown(4)."
```



```

    xfsCRDShutterOpen(2),
    xfsCRDShutterJammed(3),
    xfsCRDShutterUnknown(4),
    xfsCRDShutterNotSupported(5).\"
 ::= { xfsCRDStatusEntry 7 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.8
xfsCRDStatusGuidanceDisp OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    \"Contains the state of the guidance lights.
    xfsCRDGuidanceNotAvailable(1),
    xfsCRDGuidanceOff(2),
    xfsCRDGuidanceSlowFlash(4),
    xfsCRDGuidanceMediumFlash(8),
    xfsCRDGuidanceQuichFlash(16),
    xfsCRDGuidanceContinuous(128),
    xfsCRDGuidanceRed(256),
    xfsCRDGuidanceGreen(512),
    xfsCRDGuidanceYellow(1024),
    xfsCRDGuidanceBlue(2048),
    xfsCRDGuidanceCyan(4096),
    xfsCRDGuidanceMagenta(8192),
    xfsCRDGuidanceWhite(16384).\"
 ::= { xfsCRDStatusEntry 8 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.9
xfsCRDStatusDevicePosition OBJECT-TYPE
SYNTAX IxfsCRDDevicePositionStatus
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    \"Specifies the device position.
    xfsCRDDeviceInPosition(1),
    xfsCRDDeviceNotInPosition(2),
    xfsCRDDevicePosUnknown(3),
    xfsCRDDevicePosNotSupported(4).\"
 ::= { xfsCRDStatusEntry 9 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.10
xfsCRDStatusPowerSaveRecoveryTime 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.\"
 ::= { xfsCRDStatusEntry 10 }

-- 1.3.6.1.4.1.16213.2.14.1.2.1.100
xfsCRDStatusExtraStatus OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    \"Vendor dependent additional device status information.\"
 ::= { xfsCRDStatusEntry 100 }

-- *****
-- CRD Sub Device Status Table

```

## CWA 15748-45:2011 (E)

```
--
-- The ASN.1 prefix for Version 1 of CRD is: 1.3.6.1.4.1.16213.2.14.1.3
-- *****
-- 1.3.6.1.4.1.16213.2.14.1.3
xfsCRDSubDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsCRDSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the CRD Sub Device Status Table."
    ::= { xfsCRDV1 3 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1
xfsCRDSubDeviceEntry OBJECT-TYPE
    SYNTAX XfsCRDSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "CRD Sub-Device Status Table Entry."
    INDEX { xfsCRDSubDeviceManagedServiceName, xfsCRDSubDeviceIndex }
    ::= { xfsCRDSubDeviceTable 1 }

XfsCRDSubDeviceEntry ::=
    SEQUENCE {
        xfsCRDSubDeviceManagedServiceName
            DisplayString,
        xfsCRDSubDeviceIndex
            INTEGER,
        xfsCRDSubDeviceCUCardName
            OCTET STRING,
        xfsCRDSubDeviceCUType
            IxfsCRDCUType,
        xfsCRDSubDeviceCUInitialCount
            Integer32,
        xfsCRDSubDeviceCUCount
            Integer32,
        xfsCRDSubDeviceCUREtainCount
            Integer32,
        xfsCRDSubDeviceCUThreshold
            Integer32,
        xfsCRDSubDeviceCUSTatus
            IxfsCRDCUSTatus,
        xfsCRDSubDeviceCUHardwareSensor
            TruthValue
    }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.1
xfsCRDSubDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsCRDSubDeviceEntry 1 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.2
xfsCRDSubDeviceIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Index into the array of sub devices supported."
    ::= { xfsCRDSubDeviceEntry 2 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.3
xfsCRDSubDeviceCUCardName OBJECT-TYPE
```

```

SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The Card Unit Identifier."
 ::= { xfsCRDSubDeviceEntry 3 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.4
xfsCRDSubDeviceCUType OBJECT-TYPE
SYNTAX IxfsCRDCUType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Type of card unit.
     xfsCRDTypeSupplyBin (1),
     xfsCRDTypeRetainBin(2)."
```

```

 ::= { xfsCRDSubDeviceEntry 4 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.5
xfsCRDSubDeviceCUInitialCount OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Initial number of items contained in the card unit."
 ::= { xfsCRDSubDeviceEntry 5 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.6
xfsCRDSubDeviceCUCount OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Number of items contained in the card unit."
 ::= { xfsCRDSubDeviceEntry 6 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.7
xfsCRDSubDeviceCUREtainCount OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Number of items from this card unit contained in the retain bin."
 ::= { xfsCRDSubDeviceEntry 7 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.8
xfsCRDSubDeviceCUThreshold OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Number of items at which a threshold is fired."
 ::= { xfsCRDSubDeviceEntry 8 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.9
xfsCRDSubDeviceCUStatus OBJECT-TYPE
SYNTAX IxfsCRDCUStatus
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The status of the card unit.
     xfsCRDStatCUOK(1),
     xfsCRDStatCULow(2),
     xfsCRDStatCUEmpty(3),
```

## CWA 15748-45:2011 (E)

```

    xfsCRDStatCUInop(4),
    xfsCRDStatCUMissing(5),
    xfsCRDStatCUHigh(6),
    xfsCRDStatCUFull(7),
    xfsCRDStatCUUnknown(8)."
 ::= { xfsCRDSubDeviceEntry 9 }

-- 1.3.6.1.4.1.16213.2.14.1.3.1.10
xfsCRDSubDeviceCUHardwareSensor OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies whether or not threshold events can be generated based on hardware
sensors in the device."
    ::= { xfsCRDSubDeviceEntry 10 }

-- *****
-- CRD Error Table
-- *****
-- 1.3.6.1.4.1.16213.2.14.1.4
xfsCRDErrorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsCRDErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the CRD Error Table."
    ::= { xfsCRDV1 4 }

-- 1.3.6.1.4.1.16213.2.14.1.4.1
xfsCRDErrorEntry OBJECT-TYPE
    SYNTAX XfsCRDErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "CRD Error Table Entry."
    INDEX { xfsCRDErrorManagedServiceName, xfsCRDErrorCommandCode,
xfsCRDErrorResponseCode }
    ::= { xfsCRDErrorTable 1 }

XfsCRDErrorEntry ::=
    SEQUENCE {
        xfsCRDErrorManagedServiceName
            DisplayString,
        xfsCRDErrorCommandCode
            INTEGER,
        xfsCRDErrorResponseCode
            INTEGER,
        xfsCRDErrorCount
            Integer32
    }

-- 1.3.6.1.4.1.16213.2.14.1.4.1.1
xfsCRDErrorManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsCRDErrorEntry 1 }

-- 1.3.6.1.4.1.16213.2.14.1.4.1.2
xfsCRDErrorCommandCode OBJECT-TYPE
    SYNTAX INTEGER (1401..1500)
    MAX-ACCESS read-only
```

```

STATUS current
DESCRIPTION
  "The executable command code supported by the Service
  Provider associated with the error count of interest."
 ::= { xfsCRDErrorEntry 2 }

-- 1.3.6.1.4.1.16213.2.14.1.4.1.3
xfsCRDErrorResponseCode OBJECT-TYPE
  SYNTAX INTEGER (0..99 | 1400..1499)
  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."
 ::= { xfsCRDErrorEntry 3 }

-- 1.3.6.1.4.1.16213.2.14.1.4.1.4
xfsCRDErrorCount OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "The counter value corresponding to the managed service,
    command code and response code."
 ::= { xfsCRDErrorEntry 4 }

-- *****
-- CRD Reset Table
-- *****
-- 1.3.6.1.4.1.16213.2.14.1.5
xfsCRDResetTable OBJECT-TYPE
  SYNTAX SEQUENCE OF XfsCRDResetEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Defines the set of MIB Variables for the CRD Reset Table."
 ::= { xfsCRDV1 5 }

-- 1.3.6.1.4.1.16213.2.14.1.5.1
xfsCRDResetEntry OBJECT-TYPE
  SYNTAX XfsCRDResetEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "CRD Reset Table Entry."
  INDEX { xfsCRDResetManagedServiceName }
 ::= { xfsCRDResetTable 1 }

XfsCRDResetEntry ::=
  SEQUENCE {
    xfsCRDResetManagedServiceName
      DisplayString,
    xfsCRDResetAll
      Integer32,
    xfsCRDResetTimestamp
      DisplayString
  }

-- 1.3.6.1.4.1.16213.2.14.1.5.1.1
xfsCRDResetManagedServiceName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION

```

## CWA 15748-45:2011 (E)

```
    "Instance identifier of the managed service."
 ::= { xfsCRDResetEntry 1 }

-- 1.3.6.1.4.1.16213.2.14.1.5.1.2
xfsCRDResetAll 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."
 ::= { xfsCRDResetEntry 2 }

-- 1.3.6.1.4.1.16213.2.14.1.5.1.3
xfsCRDResetTimestamp OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Date and time the last reset of the counters was
         performed."
 ::= { xfsCRDResetEntry 3 }

-- *****
-- CRD Reset Device Table
-- *****
-- 1.3.6.1.4.1.16213.2.14.1.6
xfsCRDResetDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsCRDResetDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the CRD Reset Device Table."
 ::= { xfsCRDV1 6 }

-- 1.3.6.1.4.1.16213.2.14.1.6.1
xfsCRDResetDeviceEntry OBJECT-TYPE
    SYNTAX XfsCRDResetDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "CRD Reset Device Table Entry."
    INDEX { xfsCRDResetDeviceManagedServiceName }
 ::= { xfsCRDResetDeviceTable 1 }

XfsCRDResetDeviceEntry ::=
    SEQUENCE {
        xfsCRDResetDeviceManagedServiceName
            DisplayString,
        xfsCRDResetDeviceAction
            INTEGER,
        xfsCRDResetDeviceMediaControl
            INTEGER,
        xfsCRDResetDeviceStatus
            INTEGER
    }

-- 1.3.6.1.4.1.16213.2.14.1.6.1.1
xfsCRDResetDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
 ::= { xfsCRDResetDeviceEntry 1 }
```

```

-- 1.3.6.1.4.1.16213.2.14.1.6.1.2
xfsCRDResetDeviceAction OBJECT-TYPE
    SYNTAX INTEGER { executeReset(1) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Variable that initiates the device reset."
    ::= { xfsCRDResetDeviceEntry 2 }

-- 1.3.6.1.4.1.16213.2.14.1.6.1.3
xfsCRDResetDeviceMediaControl OBJECT-TYPE
    SYNTAX INTEGER
        {
            mediaDefault(1),
            mediaIn(2),
            mediaOut(3)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Variable that reports the media handling during the device reset."
    ::= { xfsCRDResetDeviceEntry 3 }

-- 1.3.6.1.4.1.16213.2.14.1.6.1.4
xfsCRDResetDeviceStatus OBJECT-TYPE
    SYNTAX INTEGER
        {
            resetIdle(1),
            resetInProgress(2)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Variable that reports the progress of the device reset."
    ::= { xfsCRDResetDeviceEntry 4 }

-- *****
-- CRD Device Capabilities Table
-- *****
-- 1.3.6.1.4.1.16213.2.14.1.7
xfsCRDCapabilitiesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsCRDCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the CRD capabilities table."
    ::= { xfsCRDV1 7 }

-- 1.3.6.1.4.1.16213.2.14.1.7.1
xfsCRDCapabilitiesEntry OBJECT-TYPE
    SYNTAX XfsCRDCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "CRD Device Capabilities Table Entry."
    INDEX { xfsCRDCapabilitiesManagedServiceName }
    ::= { xfsCRDCapabilitiesTable 1 }

XfsCRDCapabilitiesEntry ::=
    SEQUENCE {
        xfsCRDCapabilitiesManagedServiceName
        DisplayString,
        xfsCRDCapabilitiesCompound
    }

```

## CWA 15748-45:2011 (E)

```
    TruthValue,
    xfsCRDCapabilitiesPowerOnOption
        IxfsCRDPowerOptionCapabilities,
    xfsCRDCapabilitiesPowerOffOption
        IxfsCRDPowerOptionCapabilities,
    xfsCRDCapabilitiesCardTakenSensor
        TruthValue,
    xfsCRDCapabilitiesDispenseTo
        Integer32,
    xfsCRDCapabilitiesGuidanceCardDisp
        Integer32,
    xfsCRDCapabilitiesPowerSaveControl
        TruthValue,
    xfsCRDCapabilitiesExtraCapability
        OCTET STRING
}

-- 1.3.6.1.4.1.16213.2.14.1.7.1.1
xfsCRDCapabilitiesManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsCRDCapabilitiesEntry 1 }

-- 1.3.6.1.4.1.16213.2.14.1.7.1.2
xfsCRDCapabilitiesCompound OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Compound device capabilities.
        True(1),
        False(2)."
```



```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "Card taken sensor capability.
  True(1),
  False(2)."
```

```

::= { xfsCRDCapabilitiesEntry 5 }
```

```

-- 1.3.6.1.4.1.16213.2.14.1.7.1.6
xfsCRDCapabilitiesDispenseTo OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "Dispense to capabilities.
  xfsCRDDispenseToConsumer(1),
  xfsCRDDispenseToTransport(2)."
```

```

::= { xfsCRDCapabilitiesEntry 6 }
```

```

-- 1.3.6.1.4.1.16213.2.14.1.7.1.7
xfsCRDCapabilitiesGuidanceCardDisp OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "Contains the capability of the guidance light.
  Allowed values are a combination flag."
```

```

::= { xfsCRDCapabilitiesEntry 7 }
```

```

-- 1.3.6.1.4.1.16213.2.14.1.7.1.8
xfsCRDCapabilitiesPowerSaveControl OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "Power control capabilities.
  True(1),
  False(2)."
```

```

::= { xfsCRDCapabilitiesEntry 8 }
```

```

-- 1.3.6.1.4.1.16213.2.14.1.7.1.100
xfsCRDCapabilitiesExtraCapability OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "Vendor dependent additional device capabilities information."
```

```

::= { xfsCRDCapabilitiesEntry 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
-- *****
-- 1.3.6.1.4.1.16213.3.0.114
xfsCRDDetailedDSCTrap NOTIFICATION-TYPE
OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
```

## CWA 15748-45:2011 (E)

```
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapEvent,
        xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsCRDStatusDevice,
xfsCRDStatusNumberSubDevices, xfsCRDStatusDispenser,
        xfsCRDStatusTransport, xfsCRDStatusMedia, xfsCRDStatusShutter,
xfsCRDStatusGuidanceDisp, xfsCRDStatusDevicePosition,
        xfsCRDStatusPowerSaveRecoveryTime, xfsCRDStatusExtraStatus }
STATUS current
DESCRIPTION
    "This trap indicates a change in the status of a managed
    service."
 ::= { xfsTrapV2 114 }

-- 1.3.6.1.4.1.16213.3.0.214
xfsCRDSubDeviceTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapManagedServiceName, xfsCommonTrapManagedServiceClass,
xfsCommonTrapManagedServiceClassName, xfsCommonTrapManagedServiceType,
xfsCommonTrapManagedServiceOid,
        xfsCommonTrapPhysicalDeviceName, xfsCommonTrapDeviceVendor,
xfsCommonTrapMIBVersion, xfsCommonTrapEvent, xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsCRDSubDeviceIndex, xfsCRDSubDeviceCUCardName,
xfsCRDSubDeviceCUType, xfsCRDSubDeviceCUInitialCount,
        xfsCRDSubDeviceCUCount, xfsCRDSubDeviceCUREtainCount,
xfsCRDSubDeviceCUThreshold, xfsCRDSubDeviceCUStatus, xfsCRDSubDeviceCUHardwareSensor
    }
STATUS current
DESCRIPTION
    "This trap indicates a change in the status of sub-device within
    a managed service."
 ::= { xfsTrapV2 214 }

-- 1.3.6.1.4.1.16213.3.0.314
xfsCRDResetDeviceCompleteTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsCRDStatusDevice, xfsCRDStatusNumberSubDevices,
xfsCRDStatusDispenser,
        xfsCRDStatusTransport, xfsCRDStatusMedia, xfsCRDStatusShutter,
xfsCRDStatusGuidanceDisp, xfsCRDStatusDevicePosition,
        xfsCRDStatusPowerSaveRecoveryTime, xfsCRDStatusExtraStatus }
STATUS current
DESCRIPTION
    "This trap indicates the Reset action has complete and reports the
    state of the device after the reset."
 ::= { xfsTrapV2 314 }

END

--
-- SMIV2_xfsCRD.mib
--
```

## 5. Appendix B - C-Header files

---

### 5.1 XFSMIBCRD.H

---

```

/*****
*
* xfsmibcrd.h          CEN/XFS - MIB CRD
*
*                      Version 3.10  --  Dec 14, 2010
*
*****/

#ifndef __inc_xfsmibcrd_h
#define __inc_xfsmibcrd_h

#ifdef __cplusplus
extern "C" {
#endif

/*****
*   CRD Status #defines
*****/

enum IxfsCRDDispenserStatus
{
    xfsCRDDispCUOK                = 1,
    xfsCRDDispCUState,
    xfsCRDDispCUStop,
    xfsCRDDispCUUnknown
} xfsCRDDispenserStatus;

enum IxfsCRDTransportStatus
{
    xfsCRDTransportOK              = 1,
    xfsCRDTransportInop,
    xfsCRDTransportUnknown,
    xfsCRDTransportNotSupported
} xfsCRDTransportStatus;

enum IxfsCRDMediaStatus
{
    xfsCRDMediaPresent             = 2,
    xfsCRDMediaNotPresent,
    xfsCRDMediaJammed,
    xfsCRDMediaNotSupported,
    xfsCRDMediaUnknown,
    xfsCRDMediaExiting,
    xfsCRDMediaRetained
} xfsCRDMediaStatus;

enum IxfsCRDShutterStatus
{
    xfsCRDShutterClosed            = 1,
    xfsCRDShutterOpen,
    xfsCRDShutterJammed,
    xfsCRDShutterUnknown,
    xfsCRDShutterNotSupported
} xfsCRDShutterStatus;

enum IxfsCRDDevicePositionStatus
{
    xfsCRDDeviceInPosition         = 1,
    xfsCRDDeviceNotInPosition,

```

**CWA 15748-45:2011 (E)**

```
        xfsCRDDevicePosUnknown,
        xfsCRDDevicePosNotSupported
    } xfsCRDDevicePositionStatus;

/*****
 * CRD SubDevice #defines
 *****/

enum IxfsCRDCUType
{
    xfsCRDTypeSupplyBin          = 2,
    xfsCRDTypeRetainBin
} xfsCRDCUType;

enum IxfsCRDCUStatus
{
    xfsCRDStatCUOK              = 1,
    xfsCRDStatCULow,
    xfsCRDStatCUEmpty,
    xfsCRDStatCUInop,
    xfsCRDStatCUMissing,
    xfsCRDStatCUHigh,
    xfsCRDStatCUFull,
    xfsCRDStatCUUnknown
} xfsCRDCUStatus;

/*****
 * CRD Capabilities #defines
 *****/

enum IxfsCRDPowerOptionCapabilities
{
    xfsCRDPwrNoAction           = 2,
    xfsCRDPwrEject,
    xfsCRDPwrRetain,
    xfsCRDPwrEjectThenRetain
} xfsCRDPowerOption;

/*****
 *
 * MIB Variables for the Status Table
 *
 *****/

#define xfsCRDStatusManagedServiceName      (1)
#define xfsCRDStatusNumberSubDevices        (2)
#define xfsCRDStatusDevice                  (3)
#define xfsCRDStatusDispenser               (4)
#define xfsCRDStatusTransport               (5)
#define xfsCRDStatusMedia                   (6)
#define xfsCRDStatusShutter                 (7)
#define xfsCRDStatusGuidanceCardDisp        (8)
#define xfsCRDStatusDevicePosition          (9)
#define xfsCRDStatusPowerSaveRecoveryTime   (10)
#define xfsCRDStatusExtraStatus             (100)

/*****
 *
 * MIB Variables for the SubDevice Table
 *
 *****/

#define xfsCRDSubDeviceManagedServiceName    (1)
#define xfsCRDSubDeviceIndex                 (2)
#define xfsCRDSubDeviceCUCardName           (3)
#define xfsCRDSubDeviceCUType              (4)
#define xfsCRDSubDeviceCUInitialCount       (5)
#define xfsCRDSubDeviceCUCount             (6)
#define xfsCRDSubDeviceCUREtainCount        (7)
```

```

#define xfsCRDSubDeviceCUThreshold          (8)
#define xfsCRDSubDeviceCUStatus           (9)
#define xfsCRDSubDeviceCUHardwareSensor   (10)

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

/*****
*
*   MIB Variables for the Capabilities Table
*
*****/
#define xfsCRDCapabilitiesManagedServiceName   (1)
#define xfsCRDCapabilitiesCompound             (2)
#define xfsCRDCapabilitiesPowerOnOption        (3)
#define xfsCRDCapabilitiesPowerOffOption       (4)
#define xfsCRDCapabilitiesCardTakenSensor      (5)
#define xfsCRDCapabilitiesDispenseTo          (6)
#define xfsCRDCapabilitiesGuidanceCardDisp    (7)
#define xfsCRDCapabilitiesPowerSaveControl     (8)
#define xfsCRDCapabilitiesExtraCapability      (100)

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

```