

CEN

WORKSHOP

AGREEMENT

CWA 16926-47

February 2017

35.240.40

English version

**Extensions for Financial Services (XFS) interface specification –
Release 3.30 – Part 47: XFS MIB Device Specific Definitions –
Item Processing Module Device Class MIB Version 3.30**

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

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

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

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

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



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

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

© 2017 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.
Ref. No. CWA 16926-47:2017 D/E/F

Table of Contents

Foreword	3
1 Introduction.....	7
2 XFS IPM MIB variables	10
2.1 XFS IPM Status Table	10
2.1.1 <i>xfsIPMStatusTable: States</i>	10
2.2 XFS IPM Sub Device Table	19
2.2.1 <i>xfsIPMSubDeviceTable:</i>	19
2.3 XFS IPM Error Table	21
2.4 XFS IPM Reset Table.....	22
2.5 XFS IPM Reset Device Table.....	22
2.6 XFS IPM Capabilities Table.....	23
2.6.1 <i>xfsIPMCapabilitiesTable: Capabilities</i>	23
2.7 XFS IPM Sub Device Capabilities Table	34
2.7.1 <i>xfsIPMSubDeviceCapabilitiesTable:</i>	34
3 IPM Traps	36
3.1 IPM Detailed Device Status Change Trap	36
3.1.1 <i>IPM Detailed Device Status Change Trap Format</i>	36
3.1.2 <i>IPM Detailed Device Status Change Trap: an example</i>	39
3.2 IPM Sub-Device Status Change Trap	42
3.2.1 <i>IPM Sub-Device Status Change Trap Format</i>	43
3.2.2 <i>IPM Sub-Device Status Change Trap: an example</i>	45
3.3 IPM Reset Device Complete Trap	47
3.3.1 <i>IPM Reset Device Complete Trap Format</i>	47
3.3.2 <i>IPM Reset Device Complete: an example</i>	51
4 Appendix A - IPM MIB sub-tree	54
4.1 IPM MIB in SMIv2 and SMIv1 ASN-1 format.....	54
5 Appendix B - C-Header files	86
5.1 XFSIPM.H.....	86

Foreword

This CWA is revision 3.30 of the XFS interface specification.

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties on March 19th 2015, the constitution of which was supported by CEN following the public call for participation made on 1998-06-24. The specification is continuously reviewed and commented in the CEN/ISSS Workshop on XFS. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this revision 3.30.

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

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

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

Part 2: Service Classes Definition - Programmer's Reference

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Parts 19 - 28: Reserved for future use.

Parts 29 through 47 constitute an optional addendum to this CWA. They define the integration between the SNMP standard and the set of status and statistical information exported by the Service Providers.

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

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

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

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

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

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

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

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

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

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

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

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

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

Part 42: Reserved for future use.

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

Part 44: XFS MIB Application Management

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

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

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

Parts 48 - 60 are reserved for future use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Part 77: Item Processing Module Device Class Interface - Migration from Version 3.20 (CWA 16374) to Version 3.30 (this CWA) - Programmer's Reference

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a

complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is available online from <http://www.cen.eu/work/areas/ict/ebusiness/pages/ws-xfs.aspx>.

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

The formal process followed by the Workshop in the development of the CEN Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN-CENELEC Management Centre can be held accountable for the technical content of the CEN Workshop Agreement or possible conflict with standards or legislation. This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its members.

The final review/endorsement round for this CWA was started on 2015-01-16 and was successfully closed on 2015-03-19. The final text of this CWA was submitted to CEN for publication on 2015-06-19. The specification is continuously reviewed and commented in the CEN Workshop on XFS. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this revision 3.30.

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

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

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

Revision History:

3.10	December 14, 2010	Initial revision supporting the IPM specification in CEN XFS. 3.10
3.20	March 28, 2014	Update release to align the MIB with XFS 3.20.
3.30	February 6, 2017	Update release to align the MIB with XFS 3.30.

1 Introduction

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

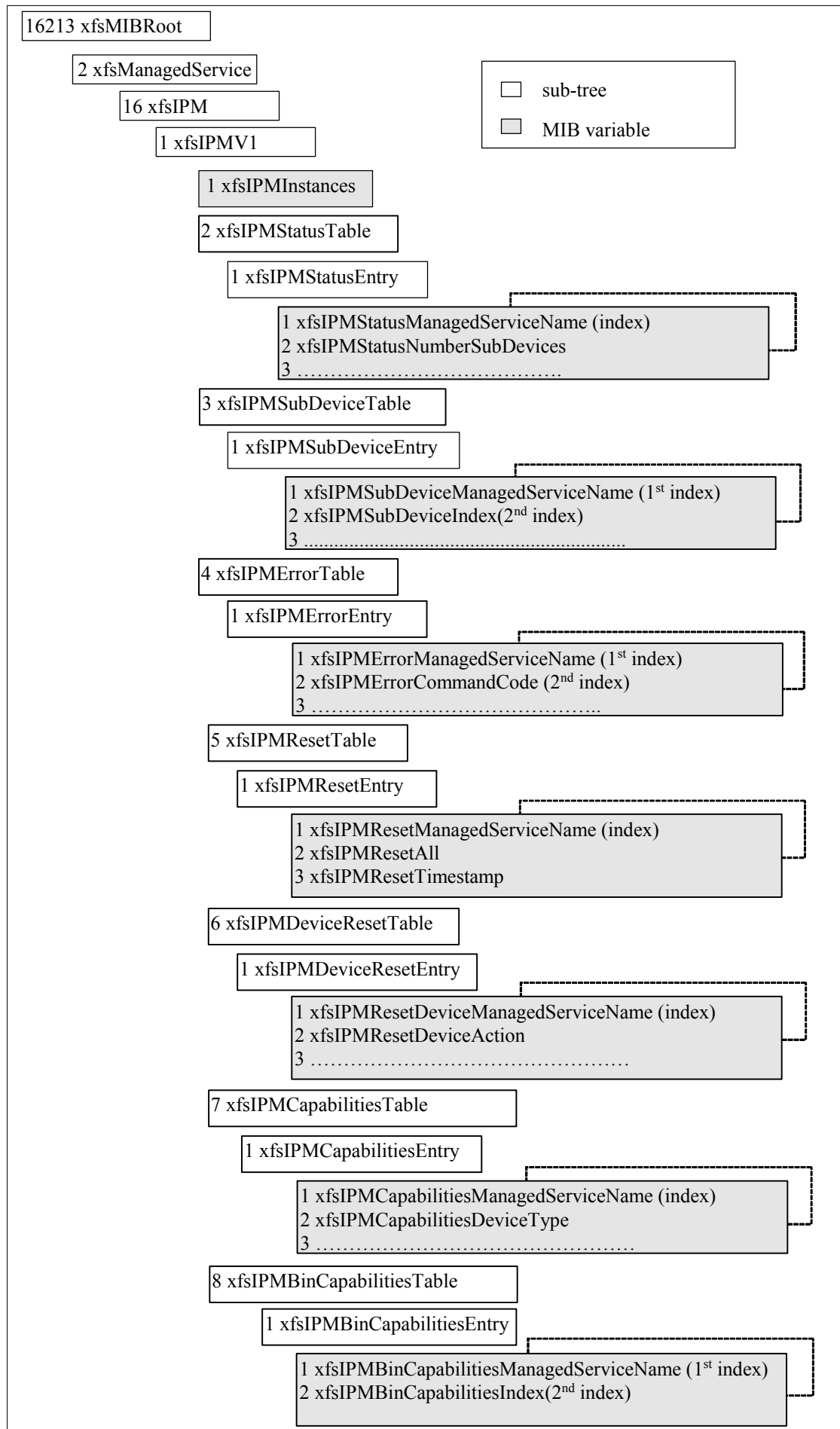
xfsMIBRoot

- xfsManagedService (2)
 - xfsIPM (16)
 - xfsIPMV1 (1)

The xfsIPMV1 sub-tree contains the following variables:

- * *xfsIPMInstances(1)* is the number of managed services for the IPM class installed on the XFS subsystem. It is a 32 bit numerical field.
- * *xfsIPMStatusTable(2)* identifies the table for the IPM variables.
- * *xfsIPMSubDeviceTable(3)* this table contains the sub-device table for the IPM device.
- * *xfsIPMErrorTable(4)* identifies the table for the IPM error counter variables.
- * *xfsIPMResetTable(5)* identifies the table for the IPM reset variable.
- * *xfsIPMResetDeviceTable(6)* identifies the table for the IPM reset device variables.
- * *xfsIPMCapabilitiesTable(7)* identifies the table for IPM capabilities variables.
- * *xfsIPMSubDeviceCapabilitiesTable(8)* identifies the table for IPM media bin 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 *xfsIPMV1* sub-tree.



Section 3 describes how the Status, Sub-Device, Error, Reset, Reset Device, and Capabilities tables apply to the IPM device class.

2 XFS IPM MIB variables

This section describes the MIB variables for the tables of the IPM Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Item Processing Module 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_IPM_RESET command to be executed from the management station.

2.1 XFS IPM Status Table

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

xfsIPMManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the IPM class.

As an example, the identifier for the device status value of *xfsIPMStatusStacker(11)* for a device with managed service name equal to "ItemProcessor1" is as follows:

Character	I	t	e	m	P	r	o	c	e	s	s	o	r	l
ASCII Hex	49	74	65	6D	50	72	6F	63	65	73	73	6F	72	31
ASCII Dec	73	116	101	109	80	114	111	99	101	115	115	111	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.2.16.1.2.1.11.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49

2.1.1 xfsIPMStatusTable: States

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

xfsIPMStatusManagedServiceName (1)

Uniquely identifies the managed service.

xfsIPMStatusNumberSubDevices (2)

Defines how many sub-devices the service has.

xfsIPMStatusDevice (3)

It contains the state of the IPM. However, an *xfsIPMStatusDevice* (3) status of *xfsDevOnline* does not necessarily imply that accepting can take place: the value of *xfsIPMStatusAcceptor* (4) field must be taken into account. Allowed values are:

Value	Meaning
xfsDevOnline(1)	The device is online (i.e. powered on and operable)
xfsDevOffline(2)	The device is offline (e.g. the operator has taken the device offline by turning a switch).
xfsDevPowerOff(3)	The device is powered off or physically not connected
xfsDevNoDevice(4)	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.
xfsDevHWEError(5)	The device is inoperable due to a hardware error
xfsDevUserError(6)	The device is present but a person is preventing proper device operation.
xfsDevBusy(7)	The device is busy and unable to process an execute command at this time.
xfsDevFraudAttempt(8)	The device is present but is inoperable because it has detected a fraud attempt.
xfsDevPotentialFraud(9)	The device is present but has detected a potential fraud attempt and is capable of remaining in service.

xfsIPMStatusAcceptor (4)

It contains the state of the overall acceptor media bins. Allowed values are:

Value	Meaning
xfsIPMAccBinOK(1)	All media bins present are in a good state
xfsIPMAccBinState (2)	One or more of the media bins is in a high, full or inoperative condition. Items can still be accepted into at least one of the media bins.
xfsIPMAccBinStop(3)	Due to a media bin problem accepting is impossible. No items can be accepted because all of the media bins are in a full or in an inoperative condition.
xfsIPMAccBinUnknown(4)	Due to a hardware error or other condition, the state of the media bins cannot be determined.

xfsIPMStatusMedia (5)

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

Value	Meaning
xfsIPMMediaPresent(1)	Media is present in the device.
xfsIPMMediaNotPresent(2)	Media is not present in the device.
xfsIPMMediaJammed(3)	Media is jammed in the device.
xfsIPMMediaNotSupported(4)	The capability to report the state of the media is not supported by the device.
xfsIPMMediaUnknown(5)	The state of the media cannot be determined with the device in its current state.
xfsIPMMediaPosition(6)	Media is at one or more of the input, output or refused positions.

xfsIPMStatusToner (6)

It contains the state of the toner or ink supply or the state of the ribbon of the endorser. Allowed values are:

Value	Meaning
xfsIPMTonerFull(1)	The toner or ink supply is full or the ribbon is OK.
xfsIPMTonerLow(2)	The toner or ink supply is low or the print contrast with a ribbon is weak.
xfsIPMTonerOut(3)	The toner or ink supply is empty or the print contrast with a ribbon is not sufficient any more.
xfsIPMTonerNotSupp(4)	The physical device does not support endorsing or the capability to report the status of the toner/ink is not supported by the device.
xfsIPMTonerUnknown(5)	Status of toner or ink supply or the ribbon cannot be determined with the device in its current state.

xfsIPMStatusInk (7)

It contains the state of the stamping ink. Allowed values are:

Value	Meaning
xfsIPMInkFull(1)	Ink supply in the device is full.
xfsIPMInkLow(2)	Ink supply in the device is low.
xfsIPMInkOut(3)	Ink supply in the device is empty.
xfsIPMInkNotSupported(4)	The physical device does not support stamping or the capability to report the status of the stamp ink supply is not supported by the device.
xfsIPMInkUnknown(5)	Status of the stamping ink supply cannot be determined with the device in its current state.

xfsIPMStatusFrontImageScanner (8)

It contains the state of the front image scanner that captures images of the front of the media items.
Allowed values are:

Value	Meaning
xfsIPMScannerOk(1)	The front scanner is OK.
xfsIPMScannerFading(2)	The front scanner performance is degraded.
xfsIPMScannerInop(3)	The front scanner is inoperable.
xfsIPMScannerNotSupported(4)	The physical device has no front scanner or the capability to report the status of the front scanner is not supported by the device.
xfsIPMScannerUnknown(5)	Status of the front scanner cannot be determined with the device in its current state.

xfsIPMStatusBackImageScanner (9)

It contains the state of the back image scanner that captures images of the back of the media items.
Allowed values are:

Value	Meaning
xfsIPMScannerOk(1)	The back scanner is OK.
xfsIPMScannerFading(2)	The back scanner performance is degraded.
xfsIPMScannerInop(3)	The back scanner is inoperable.
xfsIPMScannerNotSupported(4)	The physical device has no back scanner or the capability to report the status of the back scanner is not supported by the device.

`xfIPMScannerUnknown(5)` Status of the back scanner cannot be determined with the device in its current state.

`xfIPMStatusMICRReader (10)`

It contains the state of the MICR code line reader. Allowed values are:

Value	Meaning
<code>xfIPMMICROk(1)</code>	The MICR code line reader is OK.
<code>xfIPMMICRFading(2)</code>	The MICR code line reader performance is degraded.
<code>xfIPMMICRInop(3)</code>	The MICR code line reader is inoperative.
<code>xfIPMMICRNotSupported(4)</code>	The physical device has no MICR code line reader or the capability to report the status of the MICR code line reader is not supported by the device.
<code>xfIPMMICRUnknown(5)</code>	Status of the MICR code line reader cannot be determined with the device in its current state.

`xfIPMStatusStacker (11)`

It contains the state of the stacker (also known as an escrow). The stacker is where the media items are held while the application decides what to do with them. Allowed values are:

Value	Meaning
<code>xfIPMStackerEmpty(1)</code>	The stacker is empty.
<code>xfIPMStackerNotEmpty(2)</code>	The stacker is not empty.
<code>xfIPMStackerFull(3)</code>	The stacker is full. This state is set if the number of media items on the stacker has reached <i>usMaxMediaOnStacker</i> or some physical limit has been reached.
<code>xfIPMStackerInop(4)</code>	The stacker is inoperative.
<code>xfIPMStackerUnknown(5)</code>	Due to a hardware error or other condition, the state of the stacker cannot be determined.
<code>xfIPMStackerNotSupported(6)</code>	The physical device has no stacker or the capability to report the status of the stacker is not supported by the device.

`xfIPMStatusRebuncher (12)`

It contains the state of the rebuncher (return stacker). The rebuncher is where media items are re-bunched ready for return to the customer. Allowed values are:

Value	Meaning
<code>xfIPMRebuncherEmpty(1)</code>	The rebuncher is empty.
<code>xfIPMRebuncherNotEmpty(2)</code>	The rebuncher is not empty.
<code>xfIPMRebuncherFull(3)</code>	The rebuncher is full. This state is set if the number of media items on the rebuncher has reached its physical limit.
<code>xfIPMRebuncherInop(4)</code>	The rebuncher is inoperative.
<code>xfIPMRebuncherUnknown(5)</code>	Due to a hardware error or other condition, the state of the rebuncher cannot be determined.
<code>xfIPMRebuncherNotSupported(6)</code>	The physical device has no rebuncher or the capability to report the status of the rebuncher is not supported by the device.

`xfIPMStatusMediaFeeder (13)`

It contains the state of the media feeder. Allowed values are:

Value	Meaning
<code>xfsIPMMediaFeederEmpty(1)</code>	The media feeder is empty.
<code>xfsIPMMediaFeederNotEmpty(2)</code>	The media feeder is not empty.
<code>xfsIPMMediaFeederInop(3)</code>	The media feeder is inoperative.
<code>xfsIPMMediaFeederUnknown(4)</code>	Due to a hardware error or other condition, the state of the media feeder cannot be determined.
<code>xfsIPMMediaFeederNotSupported(5)</code>	The physical device has no media feeder or the capability to report the status of the media feeder is not supported by the device.

xfsIPMStatusShutterInput (14)

It contains the state of the shutter at the input position. Allowed values are:

Value	Meaning
<code>xfsIPMShutterClosed(1)</code>	The shutter is operational and is closed.
<code>xfsIPMShutterOpen(2)</code>	The shutter is operational and is open.
<code>xfsIPMShutterJammed(3)</code>	The shutter is jammed and is not operational. The field <i>xfsIPMStatusJammedShutterInput(33)</i> provides the positional state of the shutter.
<code>xfsIPMShutterUnknown(4)</code>	Due to a hardware error or other condition, the state of the shutter cannot be determined.
<code>xfsIPMShutterNotSupported(5)</code>	The physical device has no shutter or shutter state reporting is not supported.

xfsIPMStatusPositionInput (15)

It contains the state of the input position. Allowed values are:

Value	Meaning
<code>xfsIPMPSEmpty(1)</code>	The position is empty.
<code>xfsIPMPSNotEmpty(2)</code>	The position is not empty.
<code>xfsIPMPSUnknown(3)</code>	Due to a hardware error or other condition, the state of the position cannot be determined.
<code>xfsIPMPSNotSupported(4)</code>	The device is not capable of reporting whether or not items are at the position.

xfsIPMStatusTransportInput (16)

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

Value	Meaning
<code>xfsIPMTPOK(1)</code>	The transport is in good state.
<code>xfsIPMTPIInop(2)</code>	The transport is inoperative due to a hardware failure or media jam.
<code>xfsIPMTPIUnknown(3)</code>	Due to a hardware error or other condition, the state of the transport cannot be determined.
<code>xfsIPMTPINotSupported(4)</code>	The device either has no transport or the transport status reporting is not supported.

xfsIPMStatusTransportMediaInput (17)

It contains information regarding items which may be present on the transport at the input position. Allowed values are:

Value	Meaning
<code>xfsIPMTPMediaEmpty(1)</code>	The transport is empty.
<code>xfsIPMTPMediaNotEmpty(2)</code>	The transport is not empty.
<code>xfsIPMTPMediaUnknown(3)</code>	Due to a hardware error or other condition it is not known whether there are items on the transport.
<code>xfsIPMTPMediaNotSupported(4)</code>	The device is not capable of reporting whether or not items are on the transport.

xfsIPMStatusShutterOutput (18)

It contains the state of the shutter at the output position. Allowed values are:

Value	Meaning
<code>xfsIPMShutterClosed(1)</code>	The shutter is operational and is closed.
<code>xfsIPMShutterOpen(2)</code>	The shutter is operational and is open.
<code>xfsIPMShutterJammed(3)</code>	The shutter is jammed and is not operational. The field <i>xfsIPMStatusJammedShutterOutput(34)</i> provides the positional state of the shutter.
<code>xfsIPMShutterUnknown(4)</code>	Due to a hardware error or other condition, the state of the shutter cannot be determined.
<code>xfsIPMShutterNotSupported(5)</code>	The physical device has no shutter or shutter state reporting is not supported.

xfsIPMStatusPositionOutput (19)

It contains the state of the output position. Allowed values are:

Value	Meaning
<code>xfsIPMPSEmpty(1)</code>	The position is empty.
<code>xfsIPMPSNotEmpty(2)</code>	The position is not empty.
<code>xfsIPMPSUnknown(3)</code>	Due to a hardware error or other condition, the state of the position cannot be determined.
<code>xfsIPMPSNotSupported(4)</code>	The device is not capable of reporting whether or not items are at the position.

xfsIPMStatusTransportOutput (20)

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

Value	Meaning
<code>xfsIPMTPOK(1)</code>	The transport is in good state.
<code>xfsIPMTPInop(2)</code>	The transport is inoperative due to a hardware failure or media jam.
<code>xfsIPMTPUndknown(3)</code>	Due to a hardware error or other condition, the state of the transport cannot be determined.
<code>xfsIPMTPOutNotSupported(4)</code>	The device either has no transport or the transport status reporting is not supported.

xfsIPMStatusTransportMediaOutput (21)

It contains information regarding items which may be present on the transport at the output position. Allowed values are:

Value	Meaning
<code>xfsIPMTPMediaEmpty(1)</code>	The transport is empty.
<code>xfsIPMTPMediaNotEmpty(2)</code>	The transport is not empty.
<code>xfsIPMTPMediaUnknown(3)</code>	Due to a hardware error or other condition it is not known whether there are items on the transport.
<code>xfsIPMTPMediaNotSupported(4)</code>	The device is not capable of reporting whether or not items are on the transport.

`xfsIPMStatusShutterRefused (22)`

It contains the state of the shutter at the refused position. Allowed values are:

Value	Meaning
<code>xfsIPMShutterClosed(1)</code>	The shutter is operational and is closed.
<code>xfsIPMShutterOpen(2)</code>	The shutter is operational and is open.
<code>xfsIPMShutterJammed(3)</code>	The shutter is jammed and is not operational. The field <i><code>xfsIPMStatusJammedShutterRefused(35)</code></i> provides the positional state of the shutter.
<code>xfsIPMShutterUnknown(4)</code>	Due to a hardware error or other condition, the state of the shutter cannot be determined.
<code>xfsIPMShutterNotSupported(5)</code>	The physical device has no shutter or shutter state reporting is not supported.

`xfsIPMStatusPositionRefused (23)`

It contains the state of the refused position. Allowed values are:

Value	Meaning
<code>xfsIPMPSEmpty(1)</code>	The position is empty.
<code>xfsIPMPSNotEmpty(2)</code>	The position is not empty.
<code>xfsIPMPSUnknown(3)</code>	Due to a hardware error or other condition, the state of the position cannot be determined.
<code>xfsIPMPSNotSupported(4)</code>	The device is not capable of reporting whether or not items are at the position.

`xfsIPMStatusTransportRefused (24)`

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

Value	Meaning
<code>xfsIPMTPOK(1)</code>	The transport is in good state.
<code>xfsIPMTPInop(2)</code>	The transport is inoperative due to a hardware failure or media jam.
<code>xfsIPMTPUknown(3)</code>	Due to a hardware error or other condition, the state of the transport cannot be determined.
<code>xfsIPMTPOutSupported(4)</code>	The device either has no transport or the transport status reporting is not supported.

`xfsIPMStatusTransportMediaRefused (25)`

It contains information regarding items which may be present on the transport at the refused position. Allowed values are:

Value	Meaning
<i>xfIPMTPMediaEmpty</i> (1)	The transport is empty.
<i>xfIPMTPMediaNotEmpty</i> (2)	The transport is not empty.
<i>xfIPMTPMediaUnknown</i> (3)	Due to a hardware error or other condition it is not known whether there are items on the transport.
<i>xfIPMTPMediaNotSupported</i> (4)	The device is not capable of reporting whether or not items are on the transport.

***xfIPMStatusGuidanceMediaInput* (26)**

It contains the state of the guidance light at the input position.

Value	XFS Name	Meaning
0x00000000	WFS_IPM_GUIDANCE_NOT_AVAILABLE	The status is not available.
0x00000001	WFS_IPM_GUIDANCE_OFF	The light is turned off.
0x00000004	WFS_IPM_GUIDANCE_SLOW_FLASH	The light is blinking slowly.
0x00000008	WFS_IPM_GUIDANCE_MEDIUM_FLASH	The light is blinking medium frequency.
0x00000010	WFS_IPM_GUIDANCE_QUICK_FLASH	The light is blinking quickly.
0x00000080	WFS_IPM_GUIDANCE_CONTINUOUS	The light is turned on continuous (steady).
0x00000100	WFS_IPM_GUIDANCE_RED	The light is red.
0x00000200	WFS_IPM_GUIDANCE_GREEN	The light is green.
0x00000400	WFS_IPM_GUIDANCE_YELLOW	The light is yellow.
0x00000800	WFS_IPM_GUIDANCE_BLUE	The light is blue.
0x00001000	WFS_IPM_GUIDANCE_CYAN	The light is cyan.
0x00002000	WFS_IPM_GUIDANCE_MAGENTA	The light is magenta.
0x00004000	WFS_IPM_GUIDANCE_WHITE	The light is white.
0x00100000	WFS_IPM_GUIDANCE_ENTRY	The light is in the entry state.
0x00200000	WFS_IPM_GUIDANCE_EXIT	The light is in the exit state.

***xfIPMStatusGuidanceMediaOutput* (27)**

It contains the state of the guidance light at the output position. Allowed values are the same as variable *xfIPMStatusGuidanceMediaInput* (26).

***xfIPMStatusGuidanceMediaRefused* (28)**

It contains the state of the guidance light at the refused position. Allowed values are the same as variable *xfIPMStatusGuidanceMediaInput* (26).

***xfIPMStatusDevicePosition* (29)**

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

Value	Meaning
<i>xfIPMDeviceInPosition</i> (1)	The device is in its normal operating position, or is fixed in place and cannot be moved.
<i>xfIPMDeviceNotInPosition</i> (2)	The device has been removed from its normal operating position.
<i>xfIPMDevicePosUnknown</i> (3)	The device position cannot be determined.

xfsIPMDevicePosNotSupp(4)

The physical device does not have the capability of detecting the position.

xfsIPMStatusPowerSaveRecoveryTime (30)

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.

xfsIPMStatusMixedMode (31)

It contains whether Mixed Media mode is active. Allowed values are as follows:

Value	Meaning
xfsIPMMixedMediaNotActive(1)	Mixed Media transactions are not supported by the device or Mixed Media mode is not activated.
xfsIPMCIMMixedMedia(2)	Mixed Media mode using the CIM and IPM interfaces is activated.

xfsIPMStatusAntiFraudModule (32)

It contains the state of the anti-fraud module. Allowed values are as follows:

Value	Meaning
xfsIPMAFMNotSupported (1)	No anti-fraud module is available.
xfsIPMAFMOK(2)	Anti-fraud module is in a good state and no foreign device is detected.
xfsIPMAFMInop (3)	Anti-fraud module is inoperable.
xfsIPMAFMDeviceDetected(4)	Anti-fraud module detected the presence of a foreign device.
xfsIPMAFMUnknown(5)	The state of the anti-fraud module cannot be determined.

xfsIPMStatusJammedShutterInput (33)

It contains the state of the jammed shutter at the input position. Allowed values are:

Value	Meaning
xfsIPMShutterPositionNotSupported(1)	The physical device has no shutter or the reporting of the position of a jammed shutter is not supported.
xfsIPMShutterPositionNotJammed(2)	The shutter is not jammed.
xfsIPMShutterPositionOpen(3)	The shutter is jammed, but fully open.
xfsIPMShutterPositionPartiallyOpen(4)	The shutter is jammed, but partially open.
xfsIPMShutterPositionClosed(5)	The shutter is jammed, but fully closed.
xfsIPMShutterPositionUnknown(6)	The position of the shutter is unknown.

xfsIPMStatusJammedShutterOutput (34)

It contains the state of the jammed shutter at the output position. Allowed values are the same as variable *xfsIPMStatusJammedShutterInput(33)*.

xfsIPMStatusJammedShutterRefused (35)

It contains the state of the jammed shutter at the refused position. Allowed values are the same as variable *xfsIPMStatusJammedShutterInput(33)*.

xfsIPMStatusExtraStatus (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 IPM Sub Device Table

The *xfsIPMSubDeviceTable*(3) groups the variables identifying information for the media bins. It is indexed through two values, *xfsIPMSubDeviceManagedServiceName* and *xfsIPMSubDeviceIndex*. All sub-device table variables are read-only.

xfsIPMSubDeviceManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the IPM class. As an example, the identifier for the sub-device status value of *xfsIPMSubDeviceMBPositionName*(4) for sub-device index 1 on a device with managed service name equal to "ItemProcessor1" is as follows:

Character	I	t	e	m	P	r	o	c	e	s	s	o	r	l
ASCII Hex	49	74	65	6D	50	72	6F	63	65	73	73	6F	72	31
ASCII Dec	73	116	101	109	80	114	111	99	101	115	115	111	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.2.16.1.3.1.4.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49

2.2.1 xfsIPMSubDeviceTable:

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

It should be noted that in XFS the IPM specification uses a model of physical media bin only. The information for the IPM MIB Sub Device Table comes from the XFS IPM command WFS_INF_IPM_MEDIA_BIN_INFO.

xfsIPMSubDeviceManagedServiceName (1)
Uniquely identifies the managed service

xfsIPMSubDeviceIndex (2)
Index to the sub-device table only. This variable has no relationship to the media bin. This is an index (starting from 1) into the IPM Sub-device table.

xfsIPMSubDeviceMBBinNumber (3)
It contains the media bin number. Each structure has a unique number starting with a value of one (1) for the first structure, and incrementing by one for each subsequent structure. It is a numeric type field.

xfsIPMSubDeviceMBPositionName (4)
It contains the physical bin position name where the bin is inserted. This is an OCTET_STRING value.

xfsIPMSubDeviceMBType (5)
It contains the media bin type. It is a numeric type field and will be set to a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x00000001	WFS_IPM_TYPEMEDIAIN	Media bin. This type of bin can be specified as a destination for media items.
0x00000002	WFS_IPM_TYPERETRACT	Retract bin. This type of bin can be specified as a destination for the WFS_CMD_IPM_RETRACT_MEDI A command.

xfsIPMSubDeviceMBMediaType (6)

It contains the type of media this bin takes. It is a numeric type field. Allowed values are as follows:

Value	Meaning
<i>xfsIPMMediaTypeIPM</i> (2)	The media bin takes media through IPM device class only.
<i>xfsIPMMediaTypeCompound</i> (3)	The media bin takes media from IPM device class and from other device class (i.e. CIM).

xfsIPMSubDeviceMBBinID (7)

It contains the application defined media bin identifier. This is an OCTET STRING.

xfsIPMSubDeviceMBMediaInCount (8)

It contains the count of media that have entered the media bin as a result of operations on the IPM interface. This counter is incremented whenever media enters the media bin for any reason as a result of an operation initiated through the IPM interface. This value is persistent. On a retract-only bin, if the device cannot count media during a retract operation this value will be zero. It is a numeric type field.

xfsIPMSubDeviceMBCount (9)

It contains the total number of media in the media bin including the ones that entered via the compound device interface. If the bin is a shared bin with a compound device interface then this value may not be the same as the value of *xfsIPMSubDeviceMBMediaInCount* (8). On a retract-only bin, if the device cannot count media during a retract operation this value will be zero. It is a numeric type field.

xfsIPMSubDeviceMBRetractOperations (10)

It contains the total number of retract operations via *WFS_CMD_IPM_RETRACT_MEDIA*, *WFS_CMD_IPM_RESET* and error recovery where media is moved to the bin. This value is persistent. It is a numeric type field.

xfsIPMSubDeviceMBHardwareSensor (11)

NOTE: This field has been deprecated. The value for *bHardwareSensors* is reported using the *WFS_INF_IPM_MEDIA_BIN_CAPABILITIES* command. The below description is provided for backwards compatibility reasons only.

It contains whether or not the threshold event, *WFS_USRE_IPM_MEDIABINTHRESHOLD* (*WFS_IPM_STATMBHIGH*), can be generated based on hardware sensors in the device. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The threshold events may be generated based on hardware sensors. The threshold events will be based on the h/w sensors if the <i>xfsIPMSubDeviceMBMaximumItems</i> (12) and <i>xfsIPMSubDeviceMBMaximumRetractOperations</i> (13) are set to zero; if they are not set to zero, then the h/w sensors are ignored.
False(2)	The threshold events may not be generated based on hardware sensors.

xfsIPMSubDeviceMBMaximumItems (12)

It contains the value that controls when the threshold event *WFS_USRE_IPM_MEDIABINTHRESHOLD* (*WFS_IPM_STATMBHIGH*) will be generated; the *xfsIPMSubDeviceMBCount* (9) field must reach this value in order for the event to be generated. It is a numeric type field.

xfsIPMSubDeviceMBMaximumRetractOperations (13)

It contains the value that controls when the threshold event *WFS_USRE_IPM_MEDIABINTHRESHOLD* (*WFS_IPM_STATMBHIGH*) will be generated; the *xfsIPMSubDeviceMBRetractOperations* (10) field must reach this value in order for the event to be generated. This value is zero if the bin is not a retract bin

(i.e. does not contain the WFS_IPM_TYPERETRACT value in the *xfsIPMSubDeviceMBType* (5) field). It is a numeric type field.

xfsIPMSubDeviceMBStatus (14)

It contains the status of the media bin. Allowed values are as follows:

Value	Meaning
<i>xfsIPMMBStatOK</i> (1)	The media bin is in good state.
<i>xfsIPMMBStatFull</i> (2)	The media bin is full.
<i>xfsIPMMBStatHigh</i> (3)	The media bin is almost full (threshold)
<i>xfsIPMMBStatInop</i> (4)	The media bin is inoperative.
<i>xfsIPMMBStatMissing</i> (5)	The media bin is missing.
<i>xfsIPMMBStatUnknown</i> (6)	The media bin is unknown.
<i>xfsIPMMBStatEmpty</i> (7)	The media bin is in a good state and is empty. This is only reported where <i>xfsIPMBinCapabilitiesMBItemSensors</i> (6) = TRUE

xfsIPMSubDeviceMBExtraStatus (100)

It contains vendor dependent additional cash unit 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.

2.3 XFS IPM Error Table

The *xfsIPMErrorTable*(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

xfsIPMErrorManagedServiceName
xfsIPMErrorCommandCode
xfsIPMErrorResponseCode

The *xfsIPMErrorTable* is defined as:

- *xfsIPMErrorManagedServiceName*(1) which provides the primary index to the service in question. It is Display String field. The *xfsIPMErrorManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "ItemProcessor1".
- *xfsIPMErrorCommandCode*(2) is an index which identifies the command code that that response code is related to, e.g. WFS_CMD_IPM_MEDIA_IN (1601). It is a 32 bit numerical field.
- *xfsIPMErrorResponseCode*(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_IPM_STACKERFULL (-1603) is represented by 1603. It is a 32 bit numerical field
- *xfsIPMErrorCount*(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_IPM_STACKERFULL (-1603) error returned from the WFS_CMD_IPM_MEDIA_IN (1601) command for a device with managed service name equal to "ItemProcessor1" is as follows:

xfsMIBRoot.2.16.1.4.1.4.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49.1601.1603

2.4 XFS IPM Reset Table

The *xfsIPMResetTable(5)* contains the *xfsIPMResetAll* and *xfsIPMResetTimestamp* variables and is indexed by the single variable, *xfsIPMResetManagedServiceName*. When the *xfsIPMResetAll* 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 *xfsIPMResetTable* is defined as:

- *xfsIPMResetManagedServiceName(1)* which provides the primary index to the service in question. It is Display String field. The *xfsIPMResetManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "ItemProcessor1".
- *xfsIPMResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsIPMResetAll* 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 *xfsIPMResetAll* variable will return 0 (zero).
- *xfsIPMResetTimestamp(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 "ItemProcessor1" by setting the value zero in the *xfsIPMResetAll* variable represented by:

xfsMIBRoot.2.16.1.5.1.2.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49

2.5 XFS IPM Reset Device Table

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

The *xfsIPMResetDeviceAction* 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_IPM_RESET command will be issued.
4. Exclusive access to the device will be relinquished when the WFS_CMD_IPM_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 *xfsIPMResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request.

The *xfsIPMResetDeviceTable* is defined as:

- *xfsIPMResetDeviceManagedServiceName(1)* which provides the index to the service in question. It is a Display String field. The *xfsIPMResetDeviceManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "ItemProcessor1".

- *xfIPMResetDeviceAction(2)* is a read-write variable. Issue of a Set command on the *xfIPMResetDeviceAction* variable with value *executeReset(1)* will result in the device being reset as described above.
- *xfIPMResetDeviceMediaControl(3)* is a read only variable. This variable reports how any media found within the device is handled. The value of the *xfIPMResetDeviceMediaControl* 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. IPM retract area to retract media to) is configured through local SNMP Agent configuration.
- *xfIPMResetDeviceStatus(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 "ItemProcessor1" is reset by setting the *xfIPMResetDeviceAction* variable represented by:

xfMIBRoot.2.16.1.6.1.2.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49

2.6 XFS IPM Capabilities Table

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

xfIPMCapabilitiesManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the IPM class.

As an example, the identifier for the device status value of *xfIPMCapabilitiesDeviceType(2)* for a device with managed service name equal to "ItemProcessor1" is as follows:

Character	I	t	e	m	P	r	o	c	e	s	s	o	r	1
ASCII Hex	49	74	65	6D	50	72	6F	63	65	73	73	6F	72	31
ASCII Dec	73	116	101	109	80	114	111	99	101	115	115	111	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:

xfMIBRoot.2.16.1.7.1.2.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49

2.6.1 xfIPMCapabilitiesTable: Capabilities

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

xfIPMCapabilitiesManagedServiceName (1)

Uniquely identifies the managed service.

xfIPMCapabilitiesDeviceType (2)

It specifies the type(s) of the physical device driven by the logical service. It is a numeric type field. Allowed values are:

Value	Meaning
<i>xfIPMSingleMediaInput(2)</i>	Device accepts a single media item from customer.
<i>xfIPMBunchedMediaInput(3)</i>	Device accepts a bunch of media items from customer.

xfsIPMCapabilitiesCompoundDevice (3)

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

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

xfsIPMCapabilitiesMaxMediaOnStacker (4)

It specifies the maximum number of media items that the stacker can hold (zero if the device does not have a stacker). If the device has a bunch media input capability and the stacker is not present or has a capacity of one then the application must process each item inserted sequentially as described in section 2.2.1 of CWA 15748-18 (the Item Processing Module Device Class Interface Programmer's Reference). It is a numeric type field.

xfsIPMCapabilitiesPrintSize (5)

It specifies the size of the print area. If the media item is inserted in one of the orientations specified in *xfsIPMCapabilitiesOrientation* (22), the Service Provider will print on the back side of the media. If the media item is inserted in a different orientation to those specified in *xfsIPMCapabilitiesOrientation* (22) then printing may occur on the front side, upside down or both. It is a string type field. Allowed values are:

Value	Meaning
""	The device has no printing capabilities.
"<rows>,<cols>"	The text <rows> and <cols> are replaced with numeric values that are defined as follows: <rows> - the maximum number of rows of text that can be printed on a media item. This value is one for single line printers. <cols> - the maximum number of characters that can be printed on a row.

xfsIPMCapabilitiesStamping (6)

It specifies whether the device has stamping capability. If the media item is inserted in one of the orientations specified in *xfsIPMCapabilitiesOrientation* (22), the Service Provider will stamp on the front side of the media. If the media item is inserted in a different orientation to those specified in *xfsIPMCapabilitiesOrientation* (22) then stamping may occur on the back, upside down or both. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The capability is present.
False(2)	The capability is absent.

xfsIPMCapabilitiesRescan (7)

It specifies whether the device has the capability to either physically rescan media items after they have been inserted into the device or is able to generate any image supported by the device during the WFS_CMD_IPM_READ_IMAGE command (regardless of the images requested during the WFS_CMD_IPM_MEDIA_IN command). It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The item can be re-scanned or the images can be generated using the parameters passed in the WFS_CMD_IPM_READ_IMAGE command.

False(2)

The item cannot be re-scanned and all images required (various color, file format, bit depth) must be gathered during execution of the WFS_CMD_IPM_MEDIA_IN command.

xfsIPMCapabilitiesPresentControl (8)

It specifies how the presenting of media items is controlled during the WFS_CMD_IPM_MEDIA_IN_END and WFS_CMD_IPM_MEDIA_IN_ROLLBACK commands. This field is always set to TRUE if the device has no shutter. This field applies to all positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The presenting is controlled implicitly by the Service Provider.
False(2)	The presenting must be controlled explicitly by the application using the WFS_CMD_IPM_PRESENT_MEDIA command.

xfsIPMCapabilitiesApplicationRefuse (9)

It specifies if the Service Provider supports the WFS_CMD_IPM_MEDIA_IN mode where the application decides to accept or refuse each media item that has successfully been accepted by the device. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The Service Provider supports this mode.
False(2)	The Service Provider does not support this mode (or the device does not have a stacker).

xfsIPMCapabilitiesRetractLocations (10)

It specifies the locations to which the media can be retracted using the WFS_CMD_IPM_RETRACT_MEDIA command. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_CTRLRETRACTTOBIN	Retract the media to a retract bin.
0x00000002	WFS_IPM_CTRLRETRACTTOTRANSPORT	Retract the media to the transport.
0x00000004	WFS_IPM_CTRLRETRACTTOSTACKER	Retract the media to the stacker.
0x00000008	WFS_IPM_CTRLRETRACTTOREBUNCHER	Retract the media to the rebuncher.

xfsIPMCapabilitiesResetControl (11)

It specifies the manner in which the media can be handled on WFS_CMD_IPM_RESET. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_RESETEJECT	Eject the media.
0x00000002	WFS_IPM_RESETRERACTTOBIN	Retract the media to retract bin.
0x00000004	WFS_IPM_RESETRERACTTOTRANSPORT	Retract the media to the transport.
0x00000008	WFS_IPM_RESETRERACTTOREBUNCHER	Retract the media to the rebuncher.

xfsIPMCapabilitiesRetractCountsItems (12)

This field only applies to retract media bins. It specifies whether the bin reports the number of items retracted into the bin or just the number of retract operations. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The <i>ulCount</i> and <i>ulMediaInCount</i> include the number of media items retracted and the <i>ulMaximumItems</i> value defines when the threshold event is generated.
False(2)	The <i>ulCount</i> and <i>ulMediaInCount</i> do not contain the number of media items retracted but <i>ulRetractOperations</i> reports the number of retract operations. In this case the <i>ulMaximumRetractOperations</i> defines when the threshold event will be generated.

xfsIPMCapabilitiesImageType (13)

It specifies the image format supported by this device. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_IMAGE_TIF	The device can return scanned images in TIFF 6.0 format.
0x00000002	WFS_IPM_IMAGE_WMF	The device can return scanned images in WMF (Windows Metafile) format.
0x00000004	WFS_IPM_IMAGE_BMP	The device can return scanned images in windows BMP format.
0x00000008	WFS_IPM_IMAGE_JPG	The device can return scanned images in JPG format.

xfsIPMCapabilitiesFrontImageColorFormat (14)

It specifies the front image color formats supported by this device. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_IMAGE_COLOR_BINARY	The device can return scanned images in binary.
0x00000002	WFS_IPM_IMAGE_COLOR_GRAYSCALE	The device can return scanned images in gray scale.
0x00000004	WFS_IPM_IMAGE_COLOR_FULL	The device can return scanned images in full color.

xfsIPMCapabilitiesBackImageColorFormat (15)

It specifies the back image color formats supported by this device. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_IMAGE_COLOR_BINARY	The device can return scanned images in binary.
0x00000002	WFS_IPM_IMAGE_COLOR_GRAYSCALE	The device can return scanned images in gray scale.
0x00000004	WFS_IPM_IMAGE_COLOR_FULL	The device can return scanned images in full color.

xfsIPMCapabilitiesFrontScanColor (16)

It specifies the front image scan colors supported by this device and individually controllable by the application. Scan colors are used to enhance the scanning results on colored scan media. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_SCANCOLORRED	The device can return images scanned with red light.
0x00000002	WFS_IPM_SCANCOLORGREEN	The device can return images scanned with green light.
0x00000004	WFS_IPM_SCANCOLORBLUE	The device can return images scanned with blue light.
0x00000008	WFS_IPM_SCANCOLORYELLOW	The device can return images scanned with yellow light.
0x00000010	WFS_IPM_SCANCOLORWHITE	The device can return images scanned with white light.

`xfsIPMCapabilitiesDefaultFrontScanColor` (17)

It specifies the default front scan color format used by this device (i.e. when not explicitly set). It is a numeric type field. Allowed values are:

Value	Meaning
<code>xfsIPMScanColorRed(2)</code>	The default color is red light.
<code>xfsIPMScanColorBlue(3)</code>	The default color is blue light.
<code>xfsIPMScanColorGreen(5)</code>	The default color is green light.
<code>xfsIPMScanColorYellow(9)</code>	The default color is yellow light.
<code>xfsIPMScanColorWhite(17)</code>	The default color is white light.

`xfsIPMCapabilitiesBackScanColor` (18)

It specifies the back image scan colors supported by this device and individually controllable by the application. Scan colors are used to enhance the scanning results on colored scan media. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_SCANCOLORRED	The device can return images scanned with red light.
0x00000002	WFS_IPM_SCANCOLORGREEN	The device can return images scanned with green light.
0x00000004	WFS_IPM_SCANCOLORBLUE	The device can return images scanned with blue light.
0x00000008	WFS_IPM_SCANCOLORYELLOW	The device can return images scanned with yellow light.
0x00000010	WFS_IPM_SCANCOLORWHITE	The device can return images scanned with white light.

`xfsIPMCapabilitiesDefaultBackScanColor` (19)

It specifies the default back scan color format used by this device (i.e. when not explicitly set). It is a numeric type field. Allowed values are the same as variable *`xfsIPMCapabilitiesDefaultFrontScanColor`* (17).

xfsIPMCapabilitiesCodelineFormat (20)

It specifies the code line formats supported by this device. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_CODELINECMC7	The device can read MICR CMC7 code lines.
0x00000002	WFS_IPM_CODELINEE13B	The device can read MICR E13B code lines.
0x00000004	WFS_IPM_CODELINEOCR	The device can read code lines using Optical Character Recognition. The default or pre-configured OCR font will be used.
0x00000008	WFS_IPM_CODELINEOCRA	The device can read code lines using Optical Character Recognition font A. The ASCII codes will conform to Figure E1 in ANSI X3.17-1981.
0x00000010	WFS_IPM_CODELINEOCRB	The device can read code lines using Optical Character Recognition font B. The ASCII codes will conform to Figure C2 in ANSI X3.49-1975.

xfsIPMCapabilitiesDataSource (21)

It specifies the reading/imaging capabilities supported by this device. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_IMAGEFRONT	The device can scan the front image of the document.
0x00000002	WFS_IPM_IMAGEBACK	The device can scan the back image of the document.
0x00000004	WFS_IPM_CODELINE	The device can recognize the code line.

xfsIPMCapabilitiesOrientation (22)

It specifies the media item insertion orientations supported by the Service Provider such that hardware features such as MICR reading, endorsing and stamping will be aligned with the correct edges and sides of the media item. Devices may still return code lines and images even if one of these orientations is not used during media insertion. If the media items are inserted in one of the orientations defined in this capability then any printing or stamping will be on the correct side of the media item. If the media is inserted in a different orientation then any printing or stamping may be on the wrong side, upside down or both. This value is reported based on the customer's perspective. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_INSCODELINERIGHT	The media item should be inserted short edge first with the code line to the right.
0x00000002	WFS_IPM_INSCODELINELEFT	The media item should be inserted short edge first with the code line to the left.
0x00000004	WFS_IPM_INSCODELINEBOTTOM	The media item should be inserted long edge first with the code line to the bottom.
0x00000008	WFS_IPM_INSCODELINETOP	The media item should be inserted long edge first with the code line to the top.
0x00000010	WFS_IPM_INSFACEUP	The media item should be inserted with the front of the media item facing up.

Value	XFS Name	Meaning
0x00000020	WFS_IPM_INSFACEDOWN	The media item should be inserted with the front of the media item facing down.

xfsIPMCapabilitiesItemsTakenSensorInput (23)

It specifies whether the input position can detect when items at the exit position are taken by the user. This field relates to output and refused positions, so will always be set to FALSE for input positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
False(2)	The Service Provider does not generate an accompanying WFS_SRVE_IPM_MEDIATAKEN event when items at the exit position are taken by the user.

xfsIPMCapabilitiesItemsInsertedSensorInput (24)

It specifies whether the input position has the ability to detect when items have been inserted by the user. This field relates to all input positions, so will always be set to FALSE for output and refused positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The Service Provider generates an accompanying WFS_SRVE_IPM_MEDIINSERTED event when items have been inserted by the user.
False(2)	The Service Provider does not generate an accompanying WFS_SRVE_IPM_MEDIINSERTED event when items have been inserted by the user.

xfsIPMCapabilitiesRetractAreasInput (25)

It specifies the areas to which items may be retracted from the input position. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_CTRLRETRACTTOBIN	Can retract items in this position to a retract bin.
0x00000002	WFS_IPM_CTRLRETRACTTOTRANSPORT	Can retract items in this position to the transport.
0x00000004	WFS_IPM_CTRLRETRACTTOSTACKER	Can retract items in this position to the stacker.
0x00000008	WFS_IPM_CTRLRETRACTTOREBUNCHER	Can retract items in this position to the rebuncher.

xfsIPMCapabilitiesItemsTakenSensorOutput (26)

It specifies whether the output position can detect when items at the exit position are taken by the user. This field relates to output and refused positions, so will always be set to FALSE for input positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The Service Provider generates an accompanying WFS_SRVE_IPM_MEDIATAKEN event when items at the exit position are taken by the user.

False(2)

The Service Provider does not generate an accompanying WFS_SRVE_IPM_MEDIATAKEN event when items at the exit position are taken by the user.

xfsIPMCapabilitiesItemsInsertedSensorOutput (27)

It specifies whether the output position has the ability to detect when items have been inserted by the user. This field relates to all input positions, so will always be set to FALSE for output and refused positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
False(2)	The Service Provider does not generate an accompanying WFS_SRVE_IPM_MEDIINSERTED event when items have been inserted by the user.

xfsIPMCapabilitiesRetractAreasOutput (28)

It specifies the areas to which items may be retracted from the output position. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_CTRLRETRACTTOBIN	Can retract items in this position to a retract bin.
0x00000002	WFS_IPM_CTRLRETRACTTOTRANSPORT	Can retract items in this position to the transport.
0x00000004	WFS_IPM_CTRLRETRACTTOSTACKER	Can retract items in this position to the stacker.
0x00000008	WFS_IPM_CTRLRETRACTTOREBUNCHER	Can retract items in this position to the rebuncher.

xfsIPMCapabilitiesItemsTakenSensorRefused (29)

It specifies whether the refused position can detect when items at the exit position are taken by the user. This field relates to output and refused positions, so will always be set to FALSE for input positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The Service Provider generates an accompanying WFS_SRVE_IPM_MEDIATAKEN event when items at the exit position are taken by the user.
False(2)	The Service Provider does not generate an accompanying WFS_SRVE_IPM_MEDIATAKEN event when items at the exit position are taken by the user.

xfsIPMCapabilitiesItemsInsertedSensorRefused (30)

It specifies whether the refused position has the ability to detect when items have been inserted by the user. This field relates to all input positions, so will always be set to FALSE for output and refuse positions. It is a TruthValue type field. Allowed values are:

Value	Meaning
False(2)	The Service Provider does not generate an accompanying WFS_SRVE_IPM_MEDIINSERTED event when items have been inserted by the user.

xfsIPMCapabilitiesRetractAreasRefused (31)

It specifies the areas to which items may be retracted from the refused position. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_CTRLRETRACTTOBIN	Can retract items in this position to a retract bin.
0x00000002	WFS_IPM_CTRLRETRACTTOTRANSPORT	Can retract items in this position to the transport.
0x00000004	WFS_IPM_CTRLRETRACTTOSTACKER	Can retract items in this position to the stacker.
0x00000008	WFS_IPM_CTRLRETRACTTOREBUNCHER	Can retract items in this position to the rebuncher.

xfslPMCapabilitiesGuidanceMediaIn (32)

It specifies the capability of the media input 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_IPM_GUIDANCE_NOT_AVAILABLE	There is no guidance control available at this position.
0x00000001	WFS_IPM_GUIDANCE_OFF	The light can be off.
0x00000004	WFS_IPM_GUIDANCE_SLOW_FLASH	The light can blink slowly.
0x00000008	WFS_IPM_GUIDANCE_MEDIUM_FLASH	The light can blink medium frequency.
0x00000010	WFS_IPM_GUIDANCE_QUICK_FLASH	The light can blink quickly.
0x00000080	WFS_IPM_GUIDANCE_CONTINUOUS	The light can be continuous (steady).
0x00000100	WFS_IPM_GUIDANCE_RED	The light can be red.
0x00000200	WFS_IPM_GUIDANCE_GREEN	The light can be green.
0x00000400	WFS_IPM_GUIDANCE_YELLOW	The light can be yellow.
0x00000800	WFS_IPM_GUIDANCE_BLUE	The light can be blue.
0x00001000	WFS_IPM_GUIDANCE_CYAN	The light can be cyan.
0x00002000	WFS_IPM_GUIDANCE_MAGENTA	The light can be magenta.
0x00004000	WFS_IPM_GUIDANCE_WHITE	The light can be white.
0x00100000	WFS_IPM_GUIDANCE_ENTRY	The light is in the entry state.
0x00200000	WFS_IPM_GUIDANCE_EXIT	The light is in the exit state.

xfslPMCapabilitiesGuidanceMediaOut (33)

It specifies the capability of the media output guidelight. Allowed values are the same as variable *xfslPMCapabilitiesGuidanceMediaIn* (32).

xfslPMCapabilitiesGuidanceMediaRefused (34)

It specifies the capability of the media refused guidelight. Allowed values are the same as variable *xfslPMCapabilitiesGuidanceMediaIn* (32).

xfslPMCapabilitiesPowerSaveControl (35)

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.

xfsIPMCapabilitiesImageAfterEndorse (36)

It specifies whether the device can generate an image after text is printed on the media item. If TRUE then the generation of the image can be specified using the WFS_CMD_IPM_GET_IMAGE_AFTER_PRINT command. If FALSE, this functionality is not available. This capability applies to media items whose destination is a media bin; the *fwReturnedItemsProcessing* capability indicates whether this functionality is supported for media items that are to be returned to the customer. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	Generating an image after text is printed on the media is supported.
False(2)	Generating an image after text is printed on the media is not supported.

xfsIPMCapabilitiesReturnedItemsProcessing (37)

It specifies the processing that this device supports for media items that are identified to be returned to the customer using the WFS_CMD_IPM_SET_DESTINATION command. It is a numeric type field and will be set to a combination of hex values according to the values in the following table (zero if none of the choices is supported):

Value	XFS Name	Meaning
0x00000001	WFS_IPM_RETITEMENDORSE	This device can endorse a media item to be returned to the customer; the endorsement is specified using the WFS_CMD_IPM_PRINT_TEXT command.
0x00000002	WFS_IPM_RETITEMENDORSEIMAGE	This device can generate an image of a media item to be returned to the customer after it has been endorsed; the image is generated using the WFS_CMD_IPM_GET_IMAGE_AFTER_PRINT command.

xfsIPMCapabilitiesMixedMode (38)

It specifies whether the device supports accepting and processing items other than the types defined in the IPM specification. Allowed values are:

Value	Meaning
xfsIPMMixedMediaNotSupp (1)	Mixed Media processing is not supported.
xfsIPMCIMMixedMedia (2)	Mixed Media transactions are supported using the CIM and IPM interfaces.

xfsIPMCapabilitiesMixedDepositAndRollback (39)

It specifies whether the device can deposit one type of media and rollback the other in the same Mixed Media transaction. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The device can accept WFS_CMD_CIM_CASH_IN_END and WFS_CMD_IPM_MEDIA_IN_ROLLBACK or WFS_CMD_CIM_CASH_IN_ROLLBACK and WFS_CMD_IPM_MEDIA_IN_END to complete the current

False(2)

transaction. This value can only be True where xfsIPMCapabilitiesMixedMode is xfsIPMCIMMixedMedia. When this value is False, applications must either deposit or return ALL items to complete a transaction. Where Mixed Media transactions are not supported this value is False.

xfsIPMCapabilitiesAntiFraudModule (40)

It specifies the capability of the anti-fraud module. It is a TruthValue type field. Allowed values are:

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

xfsIPMCapabilitiesSynchronizableCommands (41)

This holds the list of execute command IDs that can be synchronized. It is an OCTET STRING with each command ID separated by a NULL terminator, while the field itself is terminated by a double NULL terminator. For example, for a device that can synchronize more than one command ID, this value will be as follows where <null> is null terminator:

commandID₁<null>commandID₂<null>.....commandID_n<null><null>

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

<null><null>

Possible values for each command ID are:

Value	XFS Name	Meaning
1601	WFS_CMD_IPM_MEDIA_IN	The WFS_CMD_IPM_MEDIA_IN command can be synchronized.
1602	WFS_CMD_IPM_MEDIA_IN_END	The WFS_CMD_IPM_MEDIA_IN_END command can be synchronized.
1603	WFS_CMD_IPM_MEDIA_IN_ROLLBACK	The WFS_CMD_IPM_MEDIA_IN_ROLLBACK command can be synchronized.
1604	WFS_CMD_IPM_READ_IMAGE	The WFS_CMD_IPM_READ_IMAGE command can be synchronized.
1605	WFS_CMD_IPM_SET_DESTINATION	The WFS_CMD_IPM_SET_DESTINATION command can be synchronized.
1606	WFS_CMD_IPM_PRESENT_MEDIA	The WFS_CMD_IPM_PRESENT_MEDIA command can be synchronized.
1607	WFS_CMD_IPM_RETRACT_MEDIA	The WFS_CMD_IPM_RETRACT_MEDIA command can be synchronized.
1608	WFS_CMD_IPM_PRINT_TEXT	The WFS_CMD_IPM_PRINT_TEXT command can be synchronized.
1609	WFS_CMD_IPM_SET_MEDIA_BIN_INFO	The WFS_CMD_IPM_SET_MEDIA_BIN_INFO command can be synchronized.
1610	WFS_CMD_IPM_RESET	The WFS_CMD_IPM_RESET command can be synchronized.

1611	WFS_CMD_IPM_SET_GUIDANCE_LIGHT	The WFS_CMD_IPM_SET_GUIDANCE_LIGHT command can be synchronized.
1612	WFS_CMD_IPM_GET_NEXT_ITEM	The WFS_CMD_IPM_GET_NEXT_ITEM command can be synchronized.
1613	WFS_CMD_IPM_ACTION_ITEM	The WFS_CMD_IPM_ACTION_ITEM command can be synchronized.
1614	WFS_CMD_IPM_EXPEL_MEDIA	The WFS_CMD_IPM_EXPEL_MEDIA command can be synchronized.
1615	WFS_CMD_IPM_GET_IMAGE_AFTER_PRINT	The WFS_CMD_IPM_GET_IMAGE_AFTER_PRINT command can be synchronized.
1616	WFS_CMD_IPM_ACCEPT_ITEM	The WFS_CMD_IPM_ACCEPT_ITEM command can be synchronized.
1617	WFS_CMD_IPM_SUPPLY_REPLENISH	The WFS_CMD_IPM_SUPPLY_REPLENISH command can be synchronized.
1618	WFS_CMD_IPM_POWER_SAVE_CONTROL	The WFS_CMD_IPM_POWER_SAVE_CONTROL command can be synchronized.
1619	WFS_CMD_IPM_SET_MODE	The WFS_CMD_IPM_SET_MODE command can be synchronized.

xfslPMCapabilitiesExtraCapability (100)

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

2.7 XFS IPM Sub Device Capabilities Table

The `xfslPMSubDeviceCapabilitiesTable(8)` groups the variables identifying information for the media bins. It is indexed through two values, `xfslPMSubDeviceCapabilitiesManagedServiceName` and `xfslPMSubDeviceCapabilitiesIndex`. All bin capabilities table variables are read-only.

`xfslPMSubDeviceCapabilitiesManagedServiceName` is the instance identifier of the managed service and uniquely identifies one instance of the IPM class. As an example, the identifier for the bin capabilities value of `xfslPMBinCapabilitiesMBPositionName(4)` for sub-device index 1 on a device with managed service name equal to "ItemProcessor1" is as follows:

Character	I	t	e	m	P	r	o	c	e	s	s	o	r	l
ASCII Hex	49	74	65	6D	50	72	6F	63	65	73	73	6F	72	31
ASCII Dec	73	116	101	109	80	114	111	99	101	115	115	111	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:

xfslMIBRoot.2.16.1.8.1.4.14.73.116.101.109.80.114.111.99.101.115.115.111.114.49

2.7.1 xfslPMSubDeviceCapabilitiesTable:

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

It should be noted that in XFS the IPM specification uses a model of physical media bin only. The information for the IPM MIB Sub Device Table comes from the XFS IPM command `WFS_INF_IPM_MEDIA_BIN_INFO`.

`xfsIPMSubDeviceCapabilitiesManagedServiceName` (1)

Uniquely identifies the managed service

`xfsIPMSubDeviceCapabilitiesIndex` (2)

Index to the bin capabilities table only. This variable has no relationship to the media bin. This is an index (starting from 1) into the IPM bin capabilities table.

`xfsIPMSubDeviceCapabilitiesMBBinNumber` (3)

It contains the media bin number. Each structure has a unique number starting with a value of one (1) for the first structure, and incrementing by one for each subsequent structure. It is a numeric type field.

`xfsIPMSubDeviceCapabilitiesMBPositionName` (4)

It contains the physical bin position name where the bin is inserted. This is an OCTET_STRING value.

`xfsIPMSubDeviceCapabilitiesMBHardwareSensors` (5)

It contains whether or not the threshold event, `WFS_USRE_IPM_MEDIABINTHRESHOLD` (`WFS_IPM_STATMBHIGH`), can be generated based on hardware sensors in the device. It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The threshold events may be generated based on hardware sensors. The threshold events will be based on the h/w sensors if the <code>xfsIPMSubDeviceMBMaximumItems</code> (12) and <code>xfsIPMSubDeviceMBMaximumRetractOperations</code> (13) are set to zero; if they are not set to zero, then the h/w sensors are ignored.
False(2)	The threshold events are not generated based on hardware sensors.

`xfsIPMSubDeviceCapabilitiesMBItemSensors` (6)

A capability that specifies whether or not the threshold event, `WFS_USRE_IPM_MEDIABININFOCHANGED` (`WFS_IPM_STATMBEMPTY`), can be generated based on hardware sensors in the device.

It is a TruthValue type field. Allowed values are:

Value	Meaning
True(1)	The threshold events can be generated and <code>xfsIPMSubDeviceMBStatus</code> (14) can report <code>xfsIPMMBStatEmpty</code> (8).
False(2)	The threshold events may not be generated based on hardware sensors.

If this value is TRUE then threshold events can be generated and `WFSIPMMEDIABIN.usStatus` can report `WFS_IPM_STATMBEMPTY`.

`xfsIPMSubDeviceCapabilitiesMBExtraCapabilities` (100)

It contains a list of vendor-specific, or any other extended, 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.

3 IPM Traps

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

3.1 IPM 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 IPM 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 IPM reflect the IPMStatus 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_IPM_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 116 defines the trap as an IPM 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 IPM Detailed Device Status Change Trap Format

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

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

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

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

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

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

This variable binding represents the XFS service class identifier generating the alarm; it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value
HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<ManagedServiceName>\class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CIM=13, IPM=16, 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_IPM_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 IPM MIB class is represented by .1.3.6.1.4.1.16213.2.16

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.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusDevice.xfsIPMStatusManagedServiceName (13)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusNumberSubDevices.xfsIPMStatusManagedServiceName (14)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusAcceptor.xfsIPMStatusManagedServiceName (15)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMedia.xfsIPMStatusManagedServiceName (16)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusToner.xfsIPMStatusManagedServiceName (17)

It contains the state of the toner or ink supply or the state of the ribbon of the endorser. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusInk.xfsIPMStatusManagedServiceName (18)

It contains the state of the stamping ink. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusFrontImageScanner.xfsIPMStatusManagedServiceName` (19)

It contains the state of the front image scanner that captures images of the front of the media items. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusBackImageScanner.xfsIPMStatusManagedServiceName` (20)

It contains the state of the back image scanner that captures images of the back of the media items. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMICRReader.xfsIPMStatusManagedServiceName` (21)

It contains the state of the MICR code line reader. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusStacker.xfsIPMStatusManagedServiceName` (22)

It contains the state of the stacker (also known as an escrow). The stacker is where the media items are held while the application decides what to do with them. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusRebuncher.xfsIPMStatusManagedServiceName` (23)

It contains the state of the rebuncher (return stacker). The rebuncher is where media items are re-bunched ready for return to the customer. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMediaFeeder.xfsIPMStatusManagedServiceName` (24)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusShutterInput.xfsIPMStatusManagedServiceName` (25)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPositionInput.xfsIPMStatusManagedServiceName` (26)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportInput.xfsIPMStatusManagedServiceName` (27)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportMediaInput.xfsIPMStatusManagedServiceName` (28)

It contains information regarding items which may be present on the transport at the input position. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusShutterOutput.xfsIPMStatusManagedServiceName` (29)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPositionOutput.xfsIPMStatusManagedServiceName` (30)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportOutput.xfsIPMStatusManagedServiceName` (31)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportMediaOutput.xfsIPMStatusManagedServiceName` (32)

It contains information regarding items which may be present on the transport at the output position. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusShutterRefused.xfsIPMStatusManagedServiceName` (33)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPositionRefused.xfsIPMStatusManagedServiceName` (34)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportRefused.xfsIPMStatusManagedServiceName (35)

It contains the state of the transport at the refused position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportMediaRefused.xfsIPMStatusManagedServiceName (36)

It contains information regarding items which may be present on the transport at the refused position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusGuidanceMediaInput.xfsIPMStatusManagedServiceName (37)

It contains the state of the guidance light indicator at the input position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusGuidanceMediaOutput.xfsIPMStatusManagedServiceName (38)

It contains the state of the guidance light indicator at the output position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusGuidanceMediaRefused.xfsIPMStatusManagedServiceName (39)

It contains the state of the guidance light indicator at the refused position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusExtraStatus.xfsIPMStatusManagedServiceName (40)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusDevicePosition.xfsIPMStatusManagedServiceName (41)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPowerSaveRecoveryTime.xfsIPMStatusManagedServiceName (42)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMixedMode.xfsIPMStatusManagedServiceName (43)

It contains whether if Mixed Media mode is active. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusAntiFraudModule.xfsIPMStatusManagedServiceName (44)

It contains the state of the anti-fraud module. It is a numeric type field.
xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusJammedShutterInput.xfsIPMStatusManagedServiceName (45)

It contains the state of the jammed shutter at the input position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusJammedShutterOutput.xfsIPMStatusManagedServiceName (46)

It contains the state of the jammed shutter at the output position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusJammedShutterRefused.xfsIPMStatusManagedServiceName (47)

It contains the state of the jammed shutter at the refused position. It is a numeric type field.

3.1.2 IPM Detailed Device Status Change Trap: an example

As an example, the following variable binding list represents a detailed device status change trap (6, 116) that is generated for an IPM with a managed service name of "ItemProcessor1". It reports that the device is OFFLINE because the Acceptor is stopped.

xfsMIBRoot.3.1.3.1	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName)
--------------------	---

	“SST System 1”
xfMIBRoot.3.1.3.2	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapManagedServiceName)
	“ItemProcessor1”
xfMIBRoot.3.1.3.3	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapManagedServiceClass)
	16 (WFS_SERVICE_CLASS_IPM)
xfMIBRoot.3.1.3.4	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapManagedServiceClassName)
	“IPM”
xfMIBRoot.3.1.3.5	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapManagedServiceType)
	2 (WFS_IPM_TYPEBUNCHMEDIAINPUT)
xfMIBRoot.3.1.3.6	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapManagedServiceOID)
	” 1.3.6.1.4.1.16213.2.16”
xfMIBRoot.3.1.3.7	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapPhysicalDeviceName)
	“ABC Corp Item Processor”
xfMIBRoot.3.1.3.8	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapDeviceVendor)
	“Best Device Incorporated”
xfMIBRoot.3.1.3.9	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapMIBVersion)
	“1.20”
xfMIBRoot.3.1.3.10	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapEvent)
	4 (WFS_SYSE_DEVICE_STATUS)
xfMIBRoot.3.1.3.11	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapDate)
	“15/05/2007 15:40:53 -300”
xfMIBRoot.3.1.3.12	(xfMIBRoot.xfTrap.xfTrapV1.xfCommonTrapVars.xfCommonTrapSPVersion)
	“3.81”
xfMIBRoot.2.16.1.2.1.3.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.xfIPMStatusDevice.xfIPMStatusManagedServiceName)
	2 (xfDevOffline)
xfMIBRoot.2.16.1.2.1.2.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.xfIPMStatusNumberSubDevices.xfIPMStatusManagedServiceName)
	1 (One sub device)
xfMIBRoot.2.16.1.2.1.4.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.xfIPMStatusAcceptor.xfIPMStatusManagedServiceName)
	3 (xfIPMAccBinStop)
xfMIBRoot.2.16.1.2.1.5.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.xfIPMStatusMedia.xfIPMStatusManagedServiceName)
	3 (xfIPMMediaJammed)
xfMIBRoot.2.16.1.2.1.6.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.xfIPMStatusToner.xfIPMStatusManagedServiceName)
	1 (xfIPMTonerFull)
xfMIBRoot.2.16.1.2.1.7.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.xfIPMStatusInk.xfIPMStatusManagedServiceName)

	3 (xfsIPMIInkOut)
xfsMIBRoot.2.16.1.2.1.8.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusFrontImageScanner .xfsIPMStatusManagedServiceName)
	1 (xfsIPMScannerOK)
xfsMIBRoot.2.16.1.2.1.9.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusBackImageScanner .xfsIPMStatusManagedServiceName)
	1 (xfsIPMScannerOK)
xfsMIBRoot.2.16.1.2.1.10.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusMICRReader .xfsIPMStatusManagedServiceName)
	1 (xfsIPMMICROK)
xfsMIBRoot.2.16.1.2.1.11.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusStacker .xfsIPMStatusManagedServiceName)
	1 (xfsIPMStackerEmpty)
xfsMIBRoot.2.16.1.2.1.12.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusRebuncher .xfsIPMStatusManagedServiceName)
	1 (xfsIPMRebuncherEmpty)
xfsMIBRoot.2.16.1.2.1.13.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusMediaFeeder .xfsIPMStatusManagedServiceName)
	1 (xfsIPMMediaFeederEmpty)
xfsMIBRoot.2.16.1.2.1.14.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusShutterInput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterClosed)
xfsMIBRoot.2.16.1.2.1.15.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusPositionInput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMPSEmpty)
xfsMIBRoot.2.16.1.2.1.16.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportInput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPOK)
xfsMIBRoot.2.16.1.2.1.17.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportMediaInput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPMediaEmpty)
xfsMIBRoot.2.16.1.2.1.18.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusShutterOutput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterClosed)
xfsMIBRoot.2.16.1.2.1.19.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusPositionOutput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMPSEmpty)
xfsMIBRoot.2.16.1.2.1.20.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportOutput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPOK)
xfsMIBRoot.2.16.1.2.1.21.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportMediaOutput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPMediaEmpty)
xfsMIBRoot.2.16.1.2.1.22.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusShutterRefused .xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterClosed)
xfsMIBRoot.2.16.1.2.1.23.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusPositionRefused .xfsIPMStatusManagedServiceName)

	1 (xfsIPMPSEmpty)
xfsMIBRoot.2.16.1.2.1.24.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportRefused .xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPOK)
xfsMIBRoot.2.16.1.2.1.25.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportMediaRefused .xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPMediaEmpty)
xfsMIBRoot.2.16.1.2.1.26.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusGuidanceMediaInput .xfsIPMStatusManagedServiceName)
	1 (value corresponding to WFS_IPM_GUIDANCE_OFF)
xfsMIBRoot.2.16.1.2.1.27.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusGuidanceMediaOutput .xfsIPMStatusManagedServiceName)
	1 (value corresponding to WFS_IPM_GUIDANCE_OFF)
xfsMIBRoot.2.16.1.2.1.28.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusGuidanceMediaRefused .xfsIPMStatusManagedServiceName)
	1 (value corresponding to WFS_IPM_GUIDANCE_OFF)
xfsMIBRoot.2.16.1.2.1.100.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusExtraStatus .xfsIPMStatusManagedServiceName)
	"\0"\0" (No extra data)
xfsMIBRoot.2.16.1.2.1.29.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusDevicePosition .xfsIPMStatusManagedServiceName)
	1 (xfsIPMDeviceInPosition)
xfsMIBRoot.2.16.1.2.1.30.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusPowerSaveRecoveryTime .xfsIPMStatusManagedServiceName)
	0 (0 seconds to return to normal operating mode)
xfsMIBRoot.2.16.1.2.1.31.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusMixedMode .xfsIPMStatusManagedServiceName)
	1 (xfsIPMMixedMediaNotActive)
xfsMIBRoot.2.16.1.2.1.32.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusAntiFraudModule .xfsIPMStatusManagedServiceName)
	2 (xfsIPMAFMOK)
xfsMIBRoot.2.16.1.2.1.33.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusJammedShutterInput .xfsIPMStatusManagedServiceName)
	3 (xfsIPMShutterPositionOpen)
xfsMIBRoot.2.16.1.2.1.34.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusJammedShutterOutput .xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterPositionNotSupported)
xfsMIBRoot.2.16.1.2.1.35.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusJammedShutterRefused .xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterPositionNotSupported)

3.2 IPM Sub-Device Status Change Trap

On the IPM device class the Sub Device Status change traps are sent when a WFS_SRVE_IPM_MEDIABININFOCHANGED 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 (fields 12-25) is defined in section [2.2](#).

The SNMP Specific trap value 216 defines the trap as an IPM Sub-Device Status Change trap.

3.2.1 IPM Sub-Device Status Change Trap Format

The following defines the variable bindings included in the IPM Sub-Device Status Change Trap.

`xfsmibRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName` (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. PTR=1, IDC=2, CIM=13, IPM=16 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_IPM_CAPABILITIES.fwType` field.

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

This variable binding represents the OID of the sub-tree within *xfsmibRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedService* 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 IPM MIB class is represented by .1.3.6.1.4.1.16213.2.16

`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 IPM this corresponds to the `WFS_SRVE_IPM_MEDIABININFOCHANGED` 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.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceIndex.xfsIPMSubDeviceIndex (12)

Index identifying the sub-device.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBBinNumber.xfsIPMSubDeviceIndex (13)

The identity of the bin, it is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBPositionName.xfsIPMSubDeviceIndex (14)

The physical bin name, it is an OCTET_STRING.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBType.xfsIPMSubDeviceIndex (15)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBMediaType.xfsIPMSubDeviceIndex (16)

It contains media type that the bin accepts. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBBinID.xfsIPMSubDeviceIndex (17)

It contains an application defined media bin identifier. It is an OCTET_STRING field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBMediaInCount.xfsIPMSubDeviceIndex (18)

It contains the number of media that have entered the media bin as a result of operations on the IPM interface. This counter is incremented whenever media enters the media bin for any reason as a result of an operation initiated through the IPM interface. This value is persistent. On a retract-only bin, if the device cannot count media during a retract operation this value will be zero. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBCount.xfsIPMSubDeviceIndex (19)

It contains the total number of media in the media bin including the ones that entered through the compound device interface. If the bin is a shared bin with a compound device interface then this value may not be the same as the value of *xfsIPMSubDeviceMBMediaInCount* (8). On a retract-only bin, if the device cannot count media during a retract operation this value will be zero. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBRetractOperations.xfsIPMSubDeviceIndex (20)

It contains the total number of retract operations performed via WFS_CMD_IPM_RETRACT_MEDIA, WFS_CMD_IPM_RESET and error recovery where media is moved to the bin. This value is persistent. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBHardwareSensor.xfsIPMSubDeviceIndex (21)

It contains whether or not the threshold event, WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH), can be generated based on hardware sensors in the device. It is a TruthValue type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBMaximumItems.xfsIPMSubDeviceIndex (22)

It contains the value that controls when the threshold event WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH) will be generated; the *xfsIPMSubDeviceMBCount* (9) field must reach this value in order for the event to be generated. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBMaximumRetractOperations.xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex (23)

It contains the value that controls when the threshold event WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH) will be generated; the *xfsIPMSubDeviceMBRetractOperations* (10) field must reach this value in order for the event to be generated. This value is zero if the bin is not a retract bin (i.e. does not contain the WFS_IPM_TYPERETRACT value in the *xfsIPMSubDeviceMBType* (5) field). It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBStatus.xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex (24)

It contains the status of the media bin. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBExtraStatus.xfsIPMStatusManagedServiceName (25)

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

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName (26)

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.

3.2.2 IPM Sub-Device Status Change Trap: an example

As an example, the following variable binding list represents an IPM sub-device status change trap (6, 216) generated from a generic XFS SST system. This trap sends an alarm to the SNMP Manager when a WFS_SRVE_IPM_MEDIABININFOCHANGED event is generated.

xfsMIBRoot.3.1.3.1	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName)
	"SST System 1"
xfsMIBRoot.3.1.3.2	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	"ItemProcessor1"
xfsMIBRoot.3.1.3.3	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	16 (WFS_SERVICE_CLASS_IPM)
xfsMIBRoot.3.1.3.4	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	"IPM"
xfsMIBRoot.3.1.3.5	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	1 (WFS_IPM_TYPERESINGLEMEDIAINPUT)
xfsMIBRoot.3.1.3.6	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOID)
	".1.3.6.1.4.1.16213.2.16"
xfsMIBRoot.3.1.3.7	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	"ABC Corp Item Processor"
xfsMIBRoot.3.1.3.8	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	"Best Device Incorporated"

xfsMIBRoot.3.1.3.9	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	"1.20"
xfsMIBRoot.3.1.3.10	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent)
	1604 (WFS_SRVE_IPM_MEDIABININFOCHANGED)
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)
	"3.81"
xfsMIBRoot.2.16.1.3.1.2.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceIndex .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	1 (Index to first sub device)
xfsMIBRoot.2.16.1.3.1.3.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBBinNumber .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	1
xfsMIBRoot.2.16.1.3.1.4.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBPositionName .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	"Retract Bin Position"
xfsMIBRoot.2.16.1.3.1.5.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBType .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	1 (WFS_IPM_TYPEMEDIAIN)
xfsMIBRoot.2.16.1.3.1.6.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBMedia Type .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	2 (xfsIPMMediaTypeIPM)
xfsMIBRoot.2.16.1.3.1.7.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBBinID .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	"Retract Bin 1"
xfsMIBRoot.2.16.1.3.1.8.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBMediaInCount .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	10000 (10000 media)
xfsMIBRoot.2.16.1.3.1.9.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBCount .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	100 (100 media)
xfsMIBRoot.2.16.1.3.1.10.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBRetractOperations .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	30 (30 Operations)
xfsMIBRoot.2.16.1.3.1.11.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBHardwareSensor .xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	1 (TRUE)
xfsMIBRoot.2.16.1.3.1.12.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry. xfsIPMSubDeviceMBMaximumItems .xfsIPMStatusManagedServiceName.xfsIPM

	SubDeviceIndex)
	0 (trigger is the hardware sensor)
xfsMIBRoot.2.16.1.3.1.13.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBmaximumRetractOperations.xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	100 (100 operations)
xfsMIBRoot.2.16.1.3.1.14.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBStatus.xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	1 (xfsIPMMBStatOK)
xfsMIBRoot.2.16.1.3.1.100.Index1.Index2	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMSubDeviceTable.xfsIPMSubDeviceEntry.xfsIPMSubDeviceMBExtraStatus.xfsIPMStatusManagedServiceName.xfsIPMSubDeviceIndex)
	"\0"\0" (No extra data)

3.3 IPM Reset Device Complete Trap

On the IPM 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 316 defines the trap as an IPM Reset Device Complete trap.

3.3.1 IPM Reset Device Complete Trap Format

The following defines the variable bindings included in the IPM 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 **xfsIPMStatusDevice** 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, IPM=16, 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_IPM_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 item processing module MIB class is represented by .1.3.6.1.4.1.16213.2.16

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.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusDevice.xfsIPMStatusManagedServiceName (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.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusNumberSubDevices.xfsIPMStatusManagedServiceName (13)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusAcceptor.xfsIPMStatusManagedServiceName (14)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMedia.xfsIPMStatusManagedServiceName (15)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusToner.xfsIPMStatusManagedServiceName (16)

It contains the state of the toner or ink supply or the state of the ribbon of the endorser. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusInk.xfsIPMStatusManagedServiceName` (17)

It contains the state of the stamping ink. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusFrontImageScanner.xfsIPMStatusManagedServiceName` (18)

It contains the state of the front image scanner that captures images of the front of the media items. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusBackImageScanner.xfsIPMStatusManagedServiceName` (19)

It contains the state of the back image scanner that captures images of the back of the media items. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMICRReader.xfsIPMStatusManagedServiceName` (20)

It contains the state of the MICR code line reader. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusStacker.xfsIPMStatusManagedServiceName` (21)

It contains the state of the stacker (also known as an escrow). The stacker is where the media items are held while the application decides what to do with them. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusRebuncher.xfsIPMStatusManagedServiceName` (22)

It contains the state of the rebuncher (return stacker). The rebuncher is where media items are re-bunched ready for return to the customer. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMediaFeeder.xfsIPMStatusManagedServiceName` (23)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusShutterInput.xfsIPMStatusManagedServiceName` (24)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPositionInput.xfsIPMStatusManagedServiceName` (25)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportInput.xfsIPMStatusManagedServiceName` (26)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportMediaInput.xfsIPMStatusManagedServiceName` (27)

It contains information regarding items which may be present on the transport at the input position. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusShutterOutput.xfsIPMStatusManagedServiceName` (28)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPositionOutput.xfsIPMStatusManagedServiceName` (29)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportOutput.xfsIPMStatusManagedServiceName` (30)

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

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportMediaOutput.xfsIPMStatusManagedServiceName` (31)

It contains information regarding items which may be present on the transport at the output position. It is a numeric type field.

`xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusShutterRefused.xfsIPMStatusManagedServiceName` (32)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPositionRefused.xfsIPMStatusManagedServiceName (33)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportRefused.xfsIPMStatusManagedServiceName (34)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusTransportMediaRefused.xfsIPMStatusManagedServiceName (35)

It contains information regarding items which may be present on the transport at the refused position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusGuidanceMediaInput.xfsIPMStatusManagedServiceName (36)

It contains the state of the guidance light indicator at the input position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusGuidanceMediaOutput.xfsIPMStatusManagedServiceName (37)

It contains the state of the guidance light indicator at the output position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusGuidanceMediaRefused.xfsIPMStatusManagedServiceName (38)

It contains the state of the guidance light indicator at the refused position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusExtraStatus.xfsIPMStatusManagedServiceName (39)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusDevicePosition.xfsIPMStatusManagedServiceName (40)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusPowerSaveRecoveryTime.xfsIPMStatusManagedServiceName (41)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMixedMode.xfsIPMStatusManagedServiceName (42)

It contains whether if Mixed Media mode is active. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusAntiFraudModule.xfsIPMStatusManagedServiceName (43)

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

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusJammedShutterInput.xfsIPMStatusManagedServiceName (44)

It contains the state of the jammed shutter at the input position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusJammedShutterOutput.xfsIPMStatusManagedServiceName (45)

It contains the state of the jammed shutter at the output position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusJammedShutterRefused.xfsIPMStatusManagedServiceName (46)

It contains the state of the jammed shutter at the refused position. It is a numeric type field.

3.3.2 IPM Reset Device Complete: an example

As an example, the following variable binding list represents a Reset Device Complete trap (6, 316) generated as a result of a request to reset the device from the remote management station. The device in question is of type single media input with a managed service name “ItemProcessor1”.

xfMIBRoot.3.1.3.13	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult)
	0 (resetExecuted)
xfMIBRoot.3.1.3.2	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	“ItemProcessor1”
xfMIBRoot.3.1.3.3	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	16 (WFS_SERVICE_CLASS_IPM)
xfMIBRoot.3.1.3.4	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	“IPM”
xfMIBRoot.3.1.3.5	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	1 (WFS_IPM_TYPESINGLEMEDIAINPUT)
xfMIBRoot.3.1.3.6	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	” 1.3.6.1.4.1.16213.2.16”
xfMIBRoot.3.1.3.7	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	“ABC Corp Item processor”
xfMIBRoot.3.1.3.8	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	“Best Device Incorporated”
xfMIBRoot.3.1.3.9	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	“1.20”
xfMIBRoot.3.1.3.11	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	“15/05/2007 15:40:53 -300”
xfMIBRoot.3.1.3.12	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	“3.81”
xfMIBRoot.2.16.1.2.1.3.Index	(xfMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusDevice.xfsIPMStatusManagedServiceName)
	1 (xfsDevOnline)
xfMIBRoot.2.16.1.2.1.2.Index	(xfMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusNumberSubDevices.xfsIPMStatusManagedServiceName)
	1 (One sub device)
xfMIBRoot.2.16.1.2.1.4.Index	(xfMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusAcceptor.xfsIPMStatusManagedServiceName)
	3 (xfsIPMAccBinStop)
xfMIBRoot.2.16.1.2.1.5.Index	(xfMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.xfsIPMStatusMedia.xfsIPMStatusManagedServiceName)
	3 (xfsIPMMediaJammed)
xfMIBRoot.2.16.1.2.	(xfMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry.

1.6.Index	xfIPMStatusToner .xfIPMStatusManagedServiceName)
	1 (xfIPMTonerFull)
xfMIBRoot.2.16.1.2. 1.7.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusInk .xfIPMStatusManagedServiceName)
	3 (xfIPMIInkOut)
xfMIBRoot.2.16.1.2. 1.8.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusFrontImageScanner .xfIPMStatusManagedServiceName)
	1 (xfIPMScannerOK)
xfMIBRoot.2.16.1.2. 1.9.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusBackImageScanner .xfIPMStatusManagedServiceName)
	1 (xfIPMScannerOK)
xfMIBRoot.2.16.1.2. 1.10.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusMICRReader .xfIPMStatusManagedServiceName)
	1 (xfIPMMICROK)
xfMIBRoot.2.16.1.2. 1.11.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusStacker .xfIPMStatusManagedServiceName)
	1 (xfIPMStackerEmpty)
xfMIBRoot.2.16.1.2. 1.12.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusRebuncher .xfIPMStatusManagedServiceName)
	1 (xfIPMRebuncherEmpty)
xfMIBRoot.2.16.1.2. 1.13.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusMediaFeeder .xfIPMStatusManagedServiceName)
	1 (xfIPMMediaFeederEmpty)
xfMIBRoot.2.16.1.2. 1.14.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusShutterInput .xfIPMStatusManagedServiceName)
	1 (xfIPMShutterClosed)
xfMIBRoot.2.16.1.2. 1.15.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusPositionInput .xfIPMStatusManagedServiceName)
	1 (xfIPMPSEEmpty)
xfMIBRoot.2.16.1.2. 1.16.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusTransportInput .xfIPMStatusManagedServiceName)
	1 (xfIPMTPOK)
xfMIBRoot.2.16.1.2. 1.17.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusTransportMediaInput .xfIPMStatusManagedServiceName)
	1 (xfIPMTTPMediaEmpty)
xfMIBRoot.2.16.1.2. 1.18.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusShutterOutput .xfIPMStatusManagedServiceName)
	1 (xfIPMShutterClosed)
xfMIBRoot.2.16.1.2. 1.19.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusPositionOutput .xfIPMStatusManagedServiceName)
	1 (xfIPMPSEEmpty)
xfMIBRoot.2.16.1.2. 1.20.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusTransportOutput .xfIPMStatusManagedServiceName)
	1 (xfIPMTPOK)
xfMIBRoot.2.16.1.2. 1.21.Index	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry. xfIPMStatusTransportMediaOutput .xfIPMStatusManagedServiceName)
	1 (xfIPMTTPMediaEmpty)
xfMIBRoot.2.16.1.2.	(xfMIBRoot.xfManagedService.xfIPM.xfIPMV1.xfIPMStatusTable.xfIPMStatusEntry.

1.22.Index	xfsIPMStatusShutterRefused.xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterClosed)
xfsMIBRoot.2.16.1.2. 1.23.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusPositionRefused.xfsIPMStatusManagedServiceName)
	1 (xfsIPMPSEEmpty)
xfsMIBRoot.2.16.1.2. 1.24.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportRefused.xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPOK)
xfsMIBRoot.2.16.1.2. 1.25.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusTransportMediaRefused.xfsIPMStatusManagedServiceName)
	1 (xfsIPMTPMediaEmpty)
xfsMIBRoot.2.16.1.2. 1.26.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusGuidanceMediaInput.xfsIPMStatusManagedServiceName)
	1 (value corresponding to WFS_IPM_GUIDANCE_OFF)
xfsMIBRoot.2.16.1.2. 1.27.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusGuidanceMediaOutput.xfsIPMStatusManagedServiceName)
	1 (value corresponding to WFS_IPM_GUIDANCE_OFF)
xfsMIBRoot.2.16.1.2. 1.28.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusGuidanceMediaRefused.xfsIPMStatusManagedServiceName)
	1 (value corresponding to WFS_IPM_GUIDANCE_OFF)
xfsMIBRoot.2.16.1.2. 1.100.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusExtraStatus.xfsIPMStatusManagedServiceName)
	"\0"\0" (No extra data)
xfsMIBRoot.2.16.1.2. 1.29.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusDevicePosition.xfsIPMStatusManagedServiceName)
	1 (xfsIPMDeviceInPosition)
xfsMIBRoot.2.16.1.2. 1.30.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusPowerSaveRecoveryTime.xfsIPMStatusManagedServiceName)
	0 (0 seconds to return to normal operating mode)
xfsMIBRoot.2.16.1.2. 1.31.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusMixedMode.xfsIPMStatusManagedServiceName)
	1 (xfsIPMMixedMediaNotActive)
xfsMIBRoot.2.16.1.2. 1.32.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusAntiFraudModule.xfsIPMStatusManagedServiceName)
	2 (xfsIPMAFMOK)
xfsMIBRoot.2.16.1.2. 1.33.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusJammedShutterInput.xfsIPMStatusManagedServiceName)
	3 (xfsIPMShutterPositionOpen)
xfsMIBRoot.2.16.1.2. 1.34.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusJammedShutterOutput.xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterPositionNotSupported)
xfsMIBRoot.2.16.1.2. 1.35.Index	(xfsMIBRoot.xfsManagedService.xfsIPM.xfsIPMV1.xfsIPMStatusTable.xfsIPMStatusEntry. xfsIPMStatusJammedShutterRefused.xfsIPMStatusManagedServiceName)
	1 (xfsIPMShutterPositionNotSupported)

4 Appendix A - IPM MIB sub-tree

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

4.1 IPM MIB in SMIv2 and SMIv1 ASN-1 format



SMIv1_xfsIPM.MIB



SMIv2_xfsIPM.MIB

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

```
-- XFS 3.30 MIB for IPM
-- Management Information Base for XFS IPM Device
--
-- The IPM Number is 16
-- The ASN.1 prefix to, and including the IPM is: 1.3.6.1.4.1.16213.2.16
--
-- *****

XFS-IPM-MIB DEFINITIONS ::= BEGIN

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

--
-- Type definitions
--

-- Type definitions
--
-- *****
-- IPM Status #defines
-- *****

    IxfsIPMStatusAcceptor ::= INTEGER
    {
        xfsIPMAccBinOK(1),
        xfsIPMAccBinState(2),
        xfsIPMAccBinStop(3),
        xfsIPMAccBinUnknown(4)
    }

    IxfsIPMStatusMedia ::= INTEGER
    {
        xfsIPMMediaPresent(1),
        xfsIPMMediaNotPresent(2),
        xfsIPMMediaJammed(3),
        xfsIPMMediaNotSupported(4),
        xfsIPMMediaUnknown(5),
        xfsIPMMediaPosition(6)
    }

    IxfsIPMStatusToner ::= INTEGER
    {
        xfsIPMTonerFull(1),
        xfsIPMTonerLow(2),
        xfsIPMTonerOut(3),
        xfsIPMTonerNotSupp(4),
```

```

    xfsIPMTonerUnknown(5)
}

IxfIPMStatusInk ::= INTEGER
{
    xfsIPMInkFull(1),
    xfsIPMInkLow(2),
    xfsIPMInkOut(3),
    xfsIPMInkNotSupported(4),
    xfsIPMInkUnknown(5)
}

IxfIPMStatusScanner ::= INTEGER
{
    xfsIPMScannerOk(1),
    xfsIPMScannerFading(2),
    xfsIPMScannerInop(3),
    xfsIPMScannerNotSupported(4),
    xfsIPMScannerUnknown(5)
}

IxfIPMStatusMICRReader ::= INTEGER
{
    xfsIPMMICROk(1),
    xfsIPMMICRFading(2),
    xfsIPMMICRInop(3),
    xfsIPMMICRNotSupported(4),
    xfsIPMMICRUnknown(5)
}

IxfIPMStatusStacker ::= INTEGER
{
    xfsIPMStackerEmpty(1),
    xfsIPMStackerNotEmpty(2),
    xfsIPMStackerFull(3),
    xfsIPMStackerInop(4),
    xfsIPMStackerUnknown(5),
    xfsIPMStackerNotSupported(6)
}

IxfIPMStatusRebuncher ::= INTEGER
{
    xfsIPMRebuncherEmpty(1),
    xfsIPMRebuncherNotEmpty(2),
    xfsIPMRebuncherFull(3),
    xfsIPMRebuncherInop(4),
    xfsIPMRebuncherUnknown(5),
    xfsIPMRebuncherNotSupported(6)
}

IxfIPMStatusMediaFeeder ::= INTEGER
{
    xfsIPMMediaFeederEmpty(1),
    xfsIPMMediaFeederNotEmpty(2),
    xfsIPMMediaFeederInop(3),
    xfsIPMMediaFeederUnknown(4),
    xfsIPMMediaFeederNotSupported(5)
}

IxfIPMStatusShutter ::= INTEGER
{
    xfsIPMShutterClosed(1),
    xfsIPMShutterOpen(2),
    xfsIPMShutterJammed(3),
    xfsIPMShutterUnknown(4),
    xfsIPMShutterNotSupported(5)
}

```

```

IxfsIPMStatusPosition ::= INTEGER
{
    xfsIPMPSEmpty(1),
    xfsIPMPSNotEmpty(2),
    xfsIPMPSUnknown(3),
    xfsIPMPSNotSupported(4)
}

IxfsIPMStatusTransport ::= INTEGER
{
    xfsIPMTPOK(1),
    xfsIPMTPInop(2),
    xfsIPMTPUknown(3),
    xfsIPMTPINotSupported(4)
}

IxfsIPMStatusTransportMedia ::= INTEGER
{
    xfsIPMTPMediaEmpty(1),
    xfsIPMTPMediaNotEmpty(2),
    xfsIPMTPMediaUnknown(3),
    xfsIPMTPMediaNotSupported(4)
}

IxfsIPMStatusDevicePosition ::= INTEGER
{
    xfsIPMDeviceInPosition(1),
    xfsIPMDeviceNotInPosition(2),
    xfsIPMDevicePosUnknown(3),
    xfsIPMDevicePosNotSupported(4)
}

IxfsIPMStatusMixedMode ::= INTEGER
{
    xfsIPMMixedMediaNotActive(1),
    xfsIPMCIMMixedMedia(2)
}

IxfsIPMStatusAntiFraudModule ::= INTEGER
{
    xfsIPMAFMNotSupported(1),
    xfsIPMAFMOK(2),
    xfsIPMAFMInop(3),
    xfsIPMAFMDeviceDetected(4),
    xfsIPMAFMUnknown(5)
}

IxfsIPMStatusJammedShutter ::= INTEGER
{
    xfsIPMShutterPositionNotSupported(1),
    xfsIPMShutterPositionNotJammed(2),
    xfsIPMShutterPositionOpen(3),
    xfsIPMShutterPositionPartiallyOpen(4),
    xfsIPMShutterPositionClosed(5),
    xfsIPMShutterPositionUnknown(6)
}

-- *****
-- IPM SubDevice #defines
-- *****
IxfsIPMSubDeviceMBMediaType ::= INTEGER
{
    xfsIPMMediaTypeIPM(2),
    xfsIPMMediaTypeCompound(3)
}

```



```

IxfsIPMSubDeviceMBStatus ::= INTEGER
{
  xfsIPMMBStatOK(1),
  xfsIPMMBStatFull(2),
  xfsIPMMBStatHigh(3),
  xfsIPMMBStatInop(4),
  xfsIPMMBStatMissing(5),
  xfsIPMMBStatUnknown(6),
  xfsIPMMBStatEmpty(7)
}

-- *****
-- IPM Capabilities #defines
-- *****
IxfsIPMCapabilitiesDeviceType ::= INTEGER
{
  xfsIPMSingleMediaInput(2),
  xfsIPMBunchedMediaInput(3)
}

IxfsIPMCapabilitiesDefaultScanColor ::= INTEGER
{
  xfsIPMScanColorRed(2),
  xfsIPMScanColorBlue(3),
  xfsIPMScanColorGreen(5),
  xfsIPMScanColorYellow(9),
  xfsIPMScanColorWhite(17)
}

IxfsIPMCapabilitiesMixedMode ::= INTEGER
{
  xfsIPMMixedMediaNotSupp(1),
  xfsIPMCIMMixedMedia(2)
}

--
-- Node definitions
--

-- Node definitions
--
-- *****
-- Version 1 of IPM MIB
--
-- The ASN.1 prefix to, and including the Version 1 of IPM is:
1.3.6.1.4.1.16213.2.16.1
--
-- *****
-- 1.3.6.1.4.1.16213.2.16.1
xfsIPMV1 OBJECT IDENTIFIER ::= { xfsIPM 1 }

-- 1.3.6.1.4.1.16213.2.16.1.1
xfsIPMInstances OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
  "Number that represents the number of IPM managed services."
  ::= { xfsIPMV1 1 }

-- *****
-- IPM Device Status Table
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.2
xfsIPMStatusTable OBJECT-TYPE

```

```

SYNTAX SEQUENCE OF XfsIPMStatusEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Define the set of MIB Variables for the IPM status table."
 ::= { xfsIPMV1 2 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1
xfsIPMStatusEntry OBJECT-TYPE
    SYNTAX XfsIPMStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IPM Device Status Table Entry."
    INDEX { xfsIPMStatusManagedServiceName }
    ::= { xfsIPMStatusTable 1 }

XfsIPMStatusEntry ::=
    SEQUENCE {
        xfsIPMStatusManagedServiceName
            DisplayString,
        xfsIPMStatusNumberSubDevices
            Integer32,
        xfsIPMStatusDevice
            IxfsMIBDeviceStatus,
        xfsIPMStatusAcceptor
            IxfsIPMStatusAcceptor,
        xfsIPMStatusMedia
            IxfsIPMStatusMedia,
        xfsIPMStatusToner
            IxfsIPMStatusToner,
        xfsIPMStatusInk
            IxfsIPMStatusInk,
        xfsIPMStatusFrontImageScanner
            IxfsIPMStatusScanner,
        xfsIPMStatusBackImageScanner
            IxfsIPMStatusScanner,
        xfsIPMStatusMICRReader
            IxfsIPMStatusMICRReader,
        xfsIPMStatusStacker
            IxfsIPMStatusStacker,
        xfsIPMStatusRebuncher
            IxfsIPMStatusRebuncher,
        xfsIPMStatusMediaFeeder
            IxfsIPMStatusMediaFeeder,
        xfsIPMStatusShutterInput
            IxfsIPMStatusShutter,
        xfsIPMStatusPositionInput
            IxfsIPMStatusPosition,
        xfsIPMStatusTransportInput
            IxfsIPMStatusTransport,
        xfsIPMStatusTransportMediaInput
            IxfsIPMStatusTransportMedia,
        xfsIPMStatusShutterOutput
            IxfsIPMStatusShutter,
        xfsIPMStatusPositionOutput
            IxfsIPMStatusPosition,
        xfsIPMStatusTransportOutput
            IxfsIPMStatusTransport,
        xfsIPMStatusTransportMediaOutput
            IxfsIPMStatusTransportMedia,
        xfsIPMStatusShutterRefused
            IxfsIPMStatusShutter,
        xfsIPMStatusPositionRefused
            IxfsIPMStatusPosition,
        xfsIPMStatusTransportRefused

```

```

        IxfsIPMStatusTransport,
xfsIPMStatusTransportMediaRefused
        IxfsIPMStatusTransportMedia,
xfsIPMStatusGuidanceMediaInput
        Integer32,
xfsIPMStatusGuidanceMediaOutput
        Integer32,
xfsIPMStatusGuidanceMediaRefused
        Integer32,
xfsIPMStatusDevicePosition
        IxfsIPMStatusDevicePosition,
xfsIPMStatusPowerSaveRecoveryTime
        Integer32,
xfsIPMStatusMixedMode
        IxfsIPMStatusMixedMode,
xfsIPMStatusAntiFraudModule
        IxfsIPMStatusAntiFraudModule,
xfsIPMStatusJammedShutterInput
        IxfsIPMStatusJammedShutter,
xfsIPMStatusJammedShutterOutput
        IxfsIPMStatusJammedShutter,
xfsIPMStatusJammedShutterRefused
        IxfsIPMStatusJammedShutter,
xfsIPMStatusExtraStatus
        OCTET STRING
    }
}

-- 1.3.6.1.4.1.16213.2.16.1.2.1.1
xfsIPMStatusManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMStatusEntry 1 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.2
xfsIPMStatusNumberSubDevices OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of sub devices supported by the IPM device."
    ::= { xfsIPMStatusEntry 2 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.3
xfsIPMStatusDevice OBJECT-TYPE
    SYNTAX IxfsMIBDeviceStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the IPM. However, an xfsIPMStatusDevice (3)
        status of xfsDevOnline does not necessarily imply that accepting can
        take place: the value of xfsIPMStatusAcceptor (4) field must be taken
        into account."
    ::= { xfsIPMStatusEntry 3 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.4
xfsIPMStatusAcceptor OBJECT-TYPE
    SYNTAX IxfsIPMStatusAcceptor
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the overall acceptor media bins.
        Allowed values are:

```

```

        xfsIPMAccBinOK(1),
        xfsIPMAccBinState(2),
        xfsIPMAccBinStop(3),
        xfsIPMAccBinUnknown(4)"
    ::= { xfsIPMStatusEntry 4 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.5
xfsIPMStatusMedia OBJECT-TYPE
    SYNTAX IxfsIPMStatusMedia
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the media. Allowed values are:
        xfsIPMMediaPresent(1),
        xfsIPMMediaNotPresent(2),
        xfsIPMMediaJammed(3),
        xfsIPMMediaNotSupported(4),
        xfsIPMMediaUnknown(5),
        xfsIPMMediaPosition(6)"
    ::= { xfsIPMStatusEntry 5 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.6
xfsIPMStatusToner OBJECT-TYPE
    SYNTAX IxfsIPMStatusToner
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the toner or ink supply or the state of the
        ribbon of the endorser. Allowed values are:
        xfsIPMTonerFull(1),
        xfsIPMTonerLow(2),
        xfsIPMTonerOut(3),
        xfsIPMTonerNotSupp(4),
        xfsIPMTonerUnknown(5)"
    ::= { xfsIPMStatusEntry 6 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.7
xfsIPMStatusInk OBJECT-TYPE
    SYNTAX IxfsIPMStatusInk
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the stamping ink. Allowed values are:
        xfsIPMInkFull(1),
        xfsIPMInkLow(2),
        xfsIPMInkOut(3),
        xfsIPMInkNotSupported(4),
        xfsIPMInkUnknown(5)"
    ::= { xfsIPMStatusEntry 7 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.8
xfsIPMStatusFrontImageScanner OBJECT-TYPE
    SYNTAX IxfsIPMStatusScanner
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the front image scanner that captures images of
        the front of the media items. Allowed values are:
        xfsIPMScannerOk(1),
        xfsIPMScannerFading(2),
        xfsIPMScannerInop(3),
        xfsIPMScannerNotSupported(4),
        xfsIPMScannerUnknown(5)"
    ::= { xfsIPMStatusEntry 8 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.2.1.9
xfsIPMStatusBackImageScanner OBJECT-TYPE
    SYNTAX IxfsIPMStatusScanner
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the back image scanner that captures images of the
        back of the media items. Allowed values are:
        xfsIPMScannerOk(1),
        xfsIPMScannerFading(2),
        xfsIPMScannerInop(3),
        xfsIPMScannerNotSupported(4),
        xfsIPMScannerUnknown(5)"
    ::= { xfsIPMStatusEntry 9 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.10
xfsIPMStatusMICRReader OBJECT-TYPE
    SYNTAX IxfsIPMStatusMICRReader
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the MICR code line reader. Allowed values are:
        xfsIPMMICROk(1),
        xfsIPMMICRFading(2),
        xfsIPMMICRInop(3),
        xfsIPMMICRNotSupported(4),
        xfsIPMMICRUnknown(5)"
    ::= { xfsIPMStatusEntry 10 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.11
xfsIPMStatusStacker OBJECT-TYPE
    SYNTAX IxfsIPMStatusStacker
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the stacker (also known as an escrow). The stacker
        is where the media items are held while the application decides what to
        do with them. Allowed values are:
        xfsIPMStackerEmpty(1),
        xfsIPMStackerNotEmpty(2),
        xfsIPMStackerFull(3),
        xfsIPMStackerInop(4),
        xfsIPMStackerUnknown(5),
        xfsIPMStackerNotSupported(6)"
    ::= { xfsIPMStatusEntry 11 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.12
xfsIPMStatusRebuncher OBJECT-TYPE
    SYNTAX IxfsIPMStatusRebuncher
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the rebuncher (return stacker). The rebuncher is
        where media items are re-bunched ready for return to the customer.
        Allowed values are:
        xfsIPMRebuncherEmpty(1),
        xfsIPMRebuncherNotEmpty(2),
        xfsIPMRebuncherFull(3),
        xfsIPMRebuncherInop(4),
        xfsIPMRebuncherUnknown(5),
        xfsIPMRebuncherNotSupported(6)"
    ::= { xfsIPMStatusEntry 12 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.2.1.13
xfsIPMStatusMediaFeeder OBJECT-TYPE
    SYNTAX IxfsIPMStatusMediaFeeder
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the media feeder. Allowed values are:
        xfsIPMMediaFeederEmpty(1),
        xfsIPMMediaFeederNotEmpty(2),
        xfsIPMMediaFeederInop(3),
        xfsIPMMediaFeederUnknown(4),
        xfsIPMMediaFeederNotSupported(5)"
    ::= { xfsIPMStatusEntry 13 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.14
xfsIPMStatusShutterInput OBJECT-TYPE
    SYNTAX IxfsIPMStatusShutter
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the shutter at the input position.
        Allowed values are:
        xfsIPMShutterClosed(1),
        xfsIPMShutterOpen(2),
        xfsIPMShutterJammed(3),
        xfsIPMShutterUnknown(4),
        xfsIPMShutterNotSupported(5)"
    ::= { xfsIPMStatusEntry 14 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.15
xfsIPMStatusPositionInput OBJECT-TYPE
    SYNTAX IxfsIPMStatusPosition
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the input position. Allowed values are:
        xfsIPMPSEmpty(1),
        xfsIPMPSNotEmpty(2),
        xfsIPMPSUnknown(3),
        xfsIPMPSNotSupported(4)"
    ::= { xfsIPMStatusEntry 15 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.16
xfsIPMStatusTransportInput OBJECT-TYPE
    SYNTAX IxfsIPMStatusTransport
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the transport mechanism at the input position.
        Allowed values are:
        xfsIPMTPOK(1),
        xfsIPMTPInop(2),
        xfsIPMTPIUnknown(3),
        xfsIPMTPINotSupported(4)"
    ::= { xfsIPMStatusEntry 16 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.17
xfsIPMStatusTransportMediaInput OBJECT-TYPE
    SYNTAX IxfsIPMStatusTransportMedia
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains information regarding items which may be present on the

```

```

        transport at the input position. Allowed values are:
        xfsIPMTPMediaEmpty(1),
        xfsIPMTPMediaNotEmpty(2),
        xfsIPMTPMediaUnknown(3),
        xfsIPMTPMediaNotSupported(4) "
 ::= { xfsIPMStatusEntry 17 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.18
xfsIPMStatusShutterOutput OBJECT-TYPE
    SYNTAX IxfsIPMStatusShutter
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the shutter at the output position.
        Allowed values are:
        xfsIPMShutterClosed(1),
        xfsIPMShutterOpen(2),
        xfsIPMShutterJammed(3),
        xfsIPMShutterUnknown(4),
        xfsIPMShutterNotSupported(5) "
 ::= { xfsIPMStatusEntry 18 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.19
xfsIPMStatusPositionOutput OBJECT-TYPE
    SYNTAX IxfsIPMStatusPosition
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the output position. Allowed values are:
        xfsIPMPSEmpty(1),
        xfsIPMPSEmpty(2),
        xfsIPMPSEmpty(3),
        xfsIPMPSEmpty(4) "
 ::= { xfsIPMStatusEntry 19 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.20
xfsIPMStatusTransportOutput OBJECT-TYPE
    SYNTAX IxfsIPMStatusTransport
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the transport mechanism at the output position.
        Allowed values are:
        xfsIPMTPOK(1),
        xfsIPMTPOK(2),
        xfsIPMTPOK(3),
        xfsIPMTPOK(4) "
 ::= { xfsIPMStatusEntry 20 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.21
xfsIPMStatusTransportMediaOutput OBJECT-TYPE
    SYNTAX IxfsIPMStatusTransportMedia
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains information regarding items which may be present on the
        transport at the output position. Allowed values are:
        xfsIPMTPMediaEmpty(1),
        xfsIPMTPMediaNotEmpty(2),
        xfsIPMTPMediaUnknown(3),
        xfsIPMTPMediaNotSupported(4) "
 ::= { xfsIPMStatusEntry 21 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.2.1.22
xfsIPMStatusShutterRefused OBJECT-TYPE
    SYNTAX IxfsIPMStatusShutter
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the shutter at the refused position.
        Allowed values are:
        xfsIPMShutterClosed(1),
        xfsIPMShutterOpen(2),
        xfsIPMShutterJammed(3),
        xfsIPMShutterUnknown(4),
        xfsIPMShutterNotSupported(5)"
    ::= { xfsIPMStatusEntry 22 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.23
xfsIPMStatusPositionRefused OBJECT-TYPE
    SYNTAX IxfsIPMStatusPosition
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the refused position. Allowed values are:
        xfsIPMPSEmpty(1),
        xfsIPMPSNotEmpty(2),
        xfsIPMPSUnknown(3),
        xfsIPMPSNotSupported(4)"
    ::= { xfsIPMStatusEntry 23 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.24
xfsIPMStatusTransportRefused OBJECT-TYPE
    SYNTAX IxfsIPMStatusTransport
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the transport mechanism at the refused position.
        Allowed values are:
        xfsIPMTPOK(1),
        xfsIPMTPInop(2),
        xfsIPMTPUnknown(3),
        xfsIPMTPNotSupported(4)"
    ::= { xfsIPMStatusEntry 24 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.25
xfsIPMStatusTransportMediaRefused OBJECT-TYPE
    SYNTAX IxfsIPMStatusTransportMedia
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains information regarding items which may be present on the
        transport media at the refused position. Allowed values are:
        xfsIPMTPMediaEmpty(1),
        xfsIPMTPMediaNotEmpty(2),
        xfsIPMTPMediaUnknown(3),
        xfsIPMTPMediaNotSupported(4)"
    ::= { xfsIPMStatusEntry 25 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.26
xfsIPMStatusGuidanceMediaInput OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the guidance light indicator at the input
        position."

```



```

::= { xfsIPMStatusEntry 26 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.27
xfsIPMStatusGuidanceMediaOutput OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the guidance light indicator at the output
        position."
    ::= { xfsIPMStatusEntry 27 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.28
xfsIPMStatusGuidanceMediaRefused OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the state of the guidance light indicator at the refused
        position."
    ::= { xfsIPMStatusEntry 28 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.29
xfsIPMStatusDevicePosition OBJECT-TYPE
    SYNTAX IxfsIPMStatusDevicePosition
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the device position information. Allowed values are:
        xfsIPMDeviceInPosition(1),
        xfsIPMDeviceNotInPosition(2),
        xfsIPMDevicePosUnknown(3),
        xfsIPMDevicePosNotSupported(4)."
    ::= { xfsIPMStatusEntry 29 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.30
xfsIPMStatusPowerSaveRecoveryTime OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the actual number of seconds required by the device to resume
        its normal operational state from the current power saving mode."
    ::= { xfsIPMStatusEntry 30 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.31
xfsIPMStatusMixedMode OBJECT-TYPE
    SYNTAX IxfsIPMStatusMixedMode
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the mixed mode status.
        xfsIPMMixedMediaNotActive(1),
        xfsIPMCIMMixedMedia(2)."
    ::= { xfsIPMStatusEntry 31 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.32
xfsIPMStatusAntiFraudModule OBJECT-TYPE
    SYNTAX IxfsIPMStatusAntiFraudModule
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

```

```

        "Specifies the anti-fraud module status.
        xfsIPMAFMNotSupported(1),
        xfsIPMAFMOK(2),
        xfsIPMAFMInop(3),
        xfsIPMAFMDeviceDetected(4),
        xfsIPMAFMUnknown(5)."
    ::= { xfsIPMStatusEntry 32 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.33
xfsIPMStatusJammedShutterInput OBJECT-TYPE
    SYNTAX IxfsIPMStatusJammedShutter
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the status of the jammed shutter at the input position.
        xfsIPMShutterPositionNotSupported(1),
        xfsIPMShutterPositionNotJammed(2),
        xfsIPMShutterPositionOpen(3),
        xfsIPMShutterPositionPartiallyOpen(4),
        xfsIPMShutterPositionClosed(5),
        xfsIPMShutterPositionUnknown(6)"
    ::= { xfsIPMStatusEntry 33 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.34
xfsIPMStatusJammedShutterOutput OBJECT-TYPE
    SYNTAX IxfsIPMStatusJammedShutter
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the status of the jammed shutter at the output position.
        xfsIPMShutterPositionNotSupported(1),
        xfsIPMShutterPositionNotJammed(2),
        xfsIPMShutterPositionOpen(3),
        xfsIPMShutterPositionPartiallyOpen(4),
        xfsIPMShutterPositionClosed(5),
        xfsIPMShutterPositionUnknown(6)"
    ::= { xfsIPMStatusEntry 34 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.35
xfsIPMStatusJammedShutterRefused OBJECT-TYPE
    SYNTAX IxfsIPMStatusJammedShutter
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies the status of the jammed shutter at the refused position.
        xfsIPMShutterPositionNotSupported(1),
        xfsIPMShutterPositionNotJammed(2),
        xfsIPMShutterPositionOpen(3),
        xfsIPMShutterPositionPartiallyOpen(4),
        xfsIPMShutterPositionClosed(5),
        xfsIPMShutterPositionUnknown(6)"
    ::= { xfsIPMStatusEntry 35 }

-- 1.3.6.1.4.1.16213.2.16.1.2.1.100
xfsIPMStatusExtraStatus OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "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."
    ::= { xfsIPMStatusEntry 100 }

-- *****

```

```

-- IPM Sub Device Status Table
--
-- The ASN.1 prefix for Version 1 of IPM is: 1.3.6.1.4.1.16213.2.16.1.3
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.3
xfsIPMSubDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIPMSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IPM status table."
    ::= { xfsIPMV1 3 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1
xfsIPMSubDeviceEntry OBJECT-TYPE
    SYNTAX XfsIPMSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IPM Sub-Device Status Table Entry."
    INDEX { xfsIPMSubDeviceManagedServiceName, xfsIPMSubDeviceIndex }
    ::= { xfsIPMSubDeviceTable 1 }

XfsIPMSubDeviceEntry ::=
    SEQUENCE {
        xfsIPMSubDeviceManagedServiceName
            DisplayString,
        xfsIPMSubDeviceIndex
            INTEGER,
        xfsIPMSubDeviceMBBinNumber
            Integer32,
        xfsIPMSubDeviceMBPositionName
            OCTET STRING,
        xfsIPMSubDeviceMBType
            Integer32,
        xfsIPMSubDeviceMBMediaType
            IxfsIPMSubDeviceMBMediaType,
        xfsIPMSubDeviceMBBinID
            OCTET STRING,
        xfsIPMSubDeviceMBMediaInCount
            Integer32,
        xfsIPMSubDeviceMBCount
            Integer32,
        xfsIPMSubDeviceMBRetractOperations
            Integer32,
        xfsIPMSubDeviceMBHardwareSensor
            TruthValue,
        xfsIPMSubDeviceMBMaximumItems
            Integer32,
        xfsIPMSubDeviceMBMaximumRetractOperations
            Integer32,
        xfsIPMSubDeviceMBStatus
            IxfsIPMSubDeviceMBStatus,
        xfsIPMSubDeviceMBExtraStatus
            OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.1
xfsIPMSubDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMSubDeviceEntry 1 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.3.1.2
xfsIPMSubDeviceIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Index into the array of sub devices supported."
    ::= { xfsIPMSubDeviceEntry 2 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.3
xfsIPMSubDeviceMBBinNumber OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the media bin number. Each structure has a unique number
        starting with a value of one (1) for the first structure, and
        incrementing by one for each subsequent structure."
    ::= { xfsIPMSubDeviceEntry 3 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.4
xfsIPMSubDeviceMBPositionName OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the physical bin position name where the bin is inserted."
    ::= { xfsIPMSubDeviceEntry 4 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.5
xfsIPMSubDeviceMBType OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the media bin type."
    ::= { xfsIPMSubDeviceEntry 5 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.6
xfsIPMSubDeviceMBMediaType OBJECT-TYPE
    SYNTAX IxfsIPMSubDeviceMBMediaType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the type of media this bin takes. Allowed values are:
        xfsIPMMediaTypeIPM(2),
        xfsIPMMediaTypeCompound(3)"
    ::= { xfsIPMSubDeviceEntry 6 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.7
xfsIPMSubDeviceMBBinID OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the application defined media bin identifier."
    ::= { xfsIPMSubDeviceEntry 7 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.8
xfsIPMSubDeviceMBMediaInCount OBJECT-TYPE
    SYNTAX Integer32

```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Contains the count of media that have entered the media bin
    as a result of operations on the IPM interface. This counter is
    incremented whenever media enters the media bin for any reason as a
    result of an operation initiated through the IPM interface. This value is
    persistent. On a retract-only bin, if the device cannot count media
    during a retract operation this value will be zero."
::= { xfsIPMSubDeviceEntry 8 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.9
xfsIPMSubDeviceMBCount OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the total number of media in the media bin including the ones
        that entered via the compound device interface. If the bin is a shared
        bin with a compound device interface then this value may not be the same
        as the value of xfsIPMSubDeviceMBMediaInCount (8). On a retract-only bin,
        if the device cannot count media during a retract operation this value
        will be zero."
    ::= { xfsIPMSubDeviceEntry 9 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.10
xfsIPMSubDeviceMBRetractOperations OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the total number of retract operations performed via
        WFS_CMD_IPM_RETRACT_MEDIA, WFS_CMD_IPM_RESET and error recovery where
        media is moved to the bin. This value is persistent."
    ::= { xfsIPMSubDeviceEntry 10 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.11
xfsIPMSubDeviceMBHardwareSensor OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        " Contains whether or not the threshold event,
        WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH), can be generated
        based on hardware sensors in the device."
    ::= { xfsIPMSubDeviceEntry 11 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.12
xfsIPMSubDeviceMBMaximumItems OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the value that controls when the threshold event
        WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH) will be generated;
        the xfsIPMSubDeviceMBCount (9) field must reach this value in order for
        the event to be generated."
    ::= { xfsIPMSubDeviceEntry 12 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.13
xfsIPMSubDeviceMBMaximumRetractOperations OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION
    "Contains the value that controls when the threshold event
    WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH) will be generated;
    the xfsIPMSubDeviceMBRetractOperations (10) field must reach this value
    in order for the event to be generated. This value is zero if the bin is
    not a retract bin (i.e. does not contain the WFS_IPM_TYPERETRACT value in
    the xfsIPMSubDeviceMBType (5) field)."
```

```

::= { xfsIPMSubDeviceEntry 13 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.14
xfsIPMSubDeviceMBStatus OBJECT-TYPE
    SYNTAX IxfsIPMSubDeviceMBStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the status of the media bin. Allowed values are:
        xfsIPMMBStatOK(1),
        xfsIPMMBStatFull(2),
        xfsIPMMBStatHigh(3),
        xfsIPMMBStatInop(4),
        xfsIPMMBStatMissing(5),
        xfsIPMMBStatUnknown(6),
        xfsIPMMBStatEmpty(7)"
    ::= { xfsIPMSubDeviceEntry 14 }

-- 1.3.6.1.4.1.16213.2.16.1.3.1.100
xfsIPMSubDeviceMBExtraStatus OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains vendor dependent additional cash unit status information.
        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."
    ::= { xfsIPMSubDeviceEntry 100 }

-- *****
-- IPM Error Table
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.4
xfsIPMErrorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIPMErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IPM Error Table."
    ::= { xfsIPMV1 4 }

-- 1.3.6.1.4.1.16213.2.16.1.4.1
xfsIPMErrorEntry OBJECT-TYPE
    SYNTAX XfsIPMErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IPM Error Table Entry."
    INDEX { xfsIPMErrorManagedServiceName, xfsIPMErrorCommandCode,
xfsIPMErrorResponseCode }
    ::= { xfsIPMErrorTable 1 }

XfsIPMErrorEntry ::=
    SEQUENCE {
```

```

        xfsIPMErrorManagedServiceName
            DisplayString,
        xfsIPMErrorCommandCode
            INTEGER,
        xfsIPMErrorResponseCode
            INTEGER,
        xfsIPMErrorCount
            Integer32
    }

-- 1.3.6.1.4.1.16213.2.16.1.4.1.1
xfsIPMErrorManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMErrorEntry 1 }

-- 1.3.6.1.4.1.16213.2.16.1.4.1.2
xfsIPMErrorCommandCode OBJECT-TYPE
    SYNTAX INTEGER (1601..1700)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The executable command code supported by the service
        provider associated with the error count of interest."
    ::= { xfsIPMErrorEntry 2 }

-- 1.3.6.1.4.1.16213.2.16.1.4.1.3
xfsIPMErrorResponseCode OBJECT-TYPE
    SYNTAX INTEGER (0..99 | 1600..1699)
    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."
    ::= { xfsIPMErrorEntry 3 }

-- 1.3.6.1.4.1.16213.2.16.1.4.1.4
xfsIPMErrorCount OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The counter value corresponding to the managed service,
        command code and response code."
    ::= { xfsIPMErrorEntry 4 }

-- *****
-- IPM Reset Table
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.5
xfsIPMResetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIPMResetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Defines the set of MIB Variables for the IPM Reset Table."
    ::= { xfsIPMV1 5 }

-- 1.3.6.1.4.1.16213.2.16.1.5.1

```

```

xfsIPMResetEntry OBJECT-TYPE
    SYNTAX XfsIPMResetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IPM Reset Table Entry."
    INDEX { xfsIPMResetManagedServiceName }
    ::= { xfsIPMResetTable 1 }

XfsIPMResetEntry ::=
    SEQUENCE {
        xfsIPMResetManagedServiceName
            DisplayString,
        xfsIPMResetAll
            Integer32,
        xfsIPMResetTimestamp
            DisplayString
    }

-- 1.3.6.1.4.1.16213.2.16.1.5.1.1
xfsIPMResetManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMResetEntry 1 }

-- 1.3.6.1.4.1.16213.2.16.1.5.1.2
xfsIPMResetAll 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."
    ::= { xfsIPMResetEntry 2 }

-- 1.3.6.1.4.1.16213.2.16.1.5.1.3
xfsIPMResetTimestamp OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Date and time the last reset of the counters was
        performed."
    ::= { xfsIPMResetEntry 3 }

-- *****
-- IPM Reset Device Table
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.6
xfsIPMResetDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIPMResetDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IPM Reset Device Table."
    ::= { xfsIPMV1 6 }

-- 1.3.6.1.4.1.16213.2.16.1.6.1
xfsIPMResetDeviceEntry OBJECT-TYPE
    SYNTAX XfsIPMResetDeviceEntry

```



```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "IPM Reset Device Table Entry"
INDEX { xfsIPMResetDeviceManagedServiceName }
 ::= { xfsIPMResetDeviceTable 1 }

XfsIPMResetDeviceEntry ::=
    SEQUENCE {
        xfsIPMResetDeviceManagedServiceName
            DisplayString,
        xfsIPMResetDeviceAction
            INTEGER,
        xfsIPMResetDeviceMediaControl
            INTEGER,
        xfsIPMResetDeviceStatus
            INTEGER
    }

-- 1.3.6.1.4.1.16213.2.16.1.6.1.1
xfsIPMResetDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMResetDeviceEntry 1 }

-- 1.3.6.1.4.1.16213.2.16.1.6.1.2
xfsIPMResetDeviceAction OBJECT-TYPE
    SYNTAX INTEGER { executeReset(1) }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Variable that initiates the device reset"
    ::= { xfsIPMResetDeviceEntry 2 }

-- 1.3.6.1.4.1.16213.2.16.1.6.1.3
xfsIPMResetDeviceMediaControl 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"
    ::= { xfsIPMResetDeviceEntry 3 }

-- 1.3.6.1.4.1.16213.2.16.1.6.1.4
xfsIPMResetDeviceStatus OBJECT-TYPE
    SYNTAX INTEGER
        {
            resetIdle(1),
            resetInProgress(2)
        }
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Variable that reports the progress of the device reset"
    ::= { xfsIPMResetDeviceEntry 4 }

```

```

-- *****
-- IPM Device Capabilities Table
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.7
xfsIPMCapabilitiesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIPMCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IPM capabilities table."
    ::= { xfsIPMV1 7 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1
xfsIPMCapabilitiesEntry OBJECT-TYPE
    SYNTAX XfsIPMCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IPM Device Capabilities Table Entry."
    INDEX { xfsIPMCapabilitiesManagedServiceName }
    ::= { xfsIPMCapabilitiesTable 1 }

XfsIPMCapabilitiesEntry ::=
    SEQUENCE {
        xfsIPMCapabilitiesManagedServiceName
            DisplayString,
        xfsIPMCapabilitiesDeviceType
            IxfsIPMCapabilitiesDeviceType,
        xfsIPMCapabilitiesCompoundDevice
            TruthValue,
        xfsIPMCapabilitiesMaxMediaOnStacker
            Integer32,
        xfsIPMCapabilitiesPrintSize
            OCTET STRING,
        xfsIPMCapabilitiesStamping
            TruthValue,
        xfsIPMCapabilitiesRescan
            TruthValue,
        xfsIPMCapabilitiesPresentControl
            TruthValue,
        xfsIPMCapabilitiesApplicationRefuse
            TruthValue,
        xfsIPMCapabilitiesRetractLocations
            Integer32,
        xfsIPMCapabilitiesResetControl
            Integer32,
        xfsIPMCapabilitiesRetractCountsItems
            TruthValue,
        xfsIPMCapabilitiesImageType
            Integer32,
        xfsIPMCapabilitiesFrontImageColorFormat
            Integer32,
        xfsIPMCapabilitiesBackImageColorFormat
            Integer32,
        xfsIPMCapabilitiesFrontScanColor
            Integer32,
        xfsIPMCapabilitiesDefaultFrontScanColor
            IxfsIPMCapabilitiesDefaultScanColor,
        xfsIPMCapabilitiesBackScanColor
            Integer32,
        xfsIPMCapabilitiesDefaultBackScanColor
            IxfsIPMCapabilitiesDefaultScanColor,
        xfsIPMCapabilitiesCodelineFormat
            Integer32,
        xfsIPMCapabilitiesDataSource
    }

```

```

        Integer32,
xfsIPMCapabilitiesOrientation
        Integer32,
xfsIPMCapabilitiesItemsTakenSensorInput
        TruthValue,
xfsIPMCapabilitiesItemsInsertedSensorInput
        TruthValue,
xfsIPMCapabilitiesRetractAreasInput
        Integer32,
xfsIPMCapabilitiesItemsTakenSensorOutput
        TruthValue,
xfsIPMCapabilitiesItemsInsertedSensorOutput
        TruthValue,
xfsIPMCapabilitiesRetractAreasOutput
        Integer32,
xfsIPMCapabilitiesItemsTakenSensorRefused
        TruthValue,
xfsIPMCapabilitiesItemsInsertedSensorRefused
        TruthValue,
xfsIPMCapabilitiesRetractAreasRefused
        Integer32,
xfsIPMCapabilitiesGuidanceMediaIn
        Integer32,
xfsIPMCapabilitiesGuidanceMediaOut
        Integer32,
xfsIPMCapabilitiesGuidanceMediaRefused
        Integer32,
xfsIPMCapabilitiesPowerSaveControl
        TruthValue,
xfsIPMCapabilitiesImageAfterEndorse
        TruthValue,
xfsIPMCapabilitiesReturnedItemsProcessing
        Integer32,
xfsIPMCapabilitiesMixedMode
        IxfsIPMCapabilitiesMixedMode,
xfsIPMCapabilitiesMixedDepositAndRollback
        TruthValue,
xfsIPMCapabilitiesAntiFraudModule
        TruthValue,
xfsIPMCapabilitiesSynchronizableCommands
        OCTET STRING,
xfsIPMCapabilitiesExtraCapability
        OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.1
xfsIPMCapabilitiesManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMCapabilitiesEntry 1 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.2
xfsIPMCapabilitiesDeviceType OBJECT-TYPE
    SYNTAX IxfsIPMCapabilitiesDeviceType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Defines the type of device of this class. Allowed values are:
        xfsIPMSingleMediaInput(2),
        xfsIPMBunchedMediaInput(3)."

```

```

xfsIPMCapabilitiesCompoundDevice OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the logical device is part of a compound physical
        device."
    ::= { xfsIPMCapabilitiesEntry 3 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.4
xfsIPMCapabilitiesMaxMediaOnStacker OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the maximum number of media items that the stacker can
        hold (zero if the device does not have a stacker). If the device has a
        bunch media input capability and the stacker is not present or has a
        capacity of one then the application must process each item inserted
        sequentially as described in section 2.2.1 of CWA 15748-18 (the Item
        Processing Module Device Class Interface Programmer's Reference)."
    ::= { xfsIPMCapabilitiesEntry 4 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.5
xfsIPMCapabilitiesPrintSize OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the size of the print area. If the media item is inserted
        in one of the orientations specified in
        xfsIPMCapabilitiesOrientation (22), the Service Provider will print on
        the back side of the media. If the media item is inserted in a different
        orientation to those specified in xfsIPMCapabilitiesOrientation (22) then
        printing may occur on the front side, upside down or both."
    ::= { xfsIPMCapabilitiesEntry 5 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.6
xfsIPMCapabilitiesStamping OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the device has stamping capability. If the media
        item is inserted in one of the orientations specified in
        xfsIPMCapabilitiesOrientation (22), the Service Provider will stamp on
        the front side of the media. If the media item is inserted in a different
        orientation to those specified in xfsIPMCapabilitiesOrientation (22) then
        stamping may occur on the back, upside down or both."
    ::= { xfsIPMCapabilitiesEntry 6 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.7
xfsIPMCapabilitiesRescan OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the device has the capability to either physically
        rescan media items after they have been inserted into the device or is
        able to generate any image supported by the device during the
        WFS_CMD_IPM_READ_IMAGE command (regardless of the images requested during
        the WFS_CMD_IPM_MEDIA_IN command)."
    ::= { xfsIPMCapabilitiesEntry 7 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.7.1.8
xfsIPMCapabilitiesPresentControl OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies how the presenting of media items is controlled during the
        WFS_CMD_IPM_MEDIA_IN_END and WFS_CMD_IPM_MEDIA_IN_ROLLBACK commands. This
        field is always set to TRUE if the device has no shutter. This field
        applies to all output positions."
    ::= { xfsIPMCapabilitiesEntry 8 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.9
xfsIPMCapabilitiesApplicationRefuse OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies if the Service Provider supports the WFS_CMD_IPM_MEDIA_IN
        mode where the application decides to accept or refuse each media item
        that has successfully been accepted by the device."
    ::= { xfsIPMCapabilitiesEntry 9 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.10
xfsIPMCapabilitiesRetractLocations OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the locations to which the media can be retracted using the
        WFS_CMD_IPM_RETRACT_MEDIA command."
    ::= { xfsIPMCapabilitiesEntry 10 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.11
xfsIPMCapabilitiesResetControl OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the manner in which the media can be handled on
        WFS_CMD_IPM_RESET."
    ::= { xfsIPMCapabilitiesEntry 11 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.12
xfsIPMCapabilitiesRetractCountsItems OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This field only applies to retract media bins. It specifies whether the
        bin reports the number of items retracted into the bin or just the number
        of retract operations."
    ::= { xfsIPMCapabilitiesEntry 12 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.13
xfsIPMCapabilitiesImageType OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the image format supported by this device."
    ::= { xfsIPMCapabilitiesEntry 13 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.7.1.14
xfsIPMCapabilitiesFrontImageColorFormat OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the front image color formats supported by this device."
    ::= { xfsIPMCapabilitiesEntry 14 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.15
xfsIPMCapabilitiesBackImageColorFormat OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the back image color formats supported by this device."
    ::= { xfsIPMCapabilitiesEntry 15 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.16
xfsIPMCapabilitiesFrontScanColor OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the front image scan colors supported by this device and
        individually controllable by the application. Scan colors are used to
        enhance the scanning results on colored scan media."
    ::= { xfsIPMCapabilitiesEntry 16 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.17
xfsIPMCapabilitiesDefaultFrontScanColor OBJECT-TYPE
    SYNTAX IxfsIPMCapabilitiesDefaultScanColor
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the default front scan color format used by this device
        (i.e. when not explicitly set)."
    ::= { xfsIPMCapabilitiesEntry 17 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.18
xfsIPMCapabilitiesBackScanColor OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the back image scan colors supported by this device and
        individually controllable by the application. Scan colors are used to
        enhance the scanning results on colored scan media."
    ::= { xfsIPMCapabilitiesEntry 18 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.19
xfsIPMCapabilitiesDefaultBackScanColor OBJECT-TYPE
    SYNTAX IxfsIPMCapabilitiesDefaultScanColor
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the default back scan color format used by this device
        (i.e. when not explicitly set)."
    ::= { xfsIPMCapabilitiesEntry 19 }

```

```

-- 1.3.6.1.4.1.16213.2.16.1.7.1.20
xfsIPMCapabilitiesCodeLineFormat OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the code line formats supported by this device."
    ::= { xfsIPMCapabilitiesEntry 20 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.21
xfsIPMCapabilitiesDataSource OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the reading/imaging capabilities supported by this device."
    ::= { xfsIPMCapabilitiesEntry 21 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.22
xfsIPMCapabilitiesOrientation OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the media item insertion orientations supported by the
        Service Provider such that hardware features such as MICR reading,
        endorsing and stamping will be aligned with the correct edges and sides
        of the media item. Devices may still return code lines and images even if
        one of these orientations is not used during media insertion. If the
        media items are inserted in one of the orientations defined in this
        capability then any printing or stamping will be on the correct side of
        the media item. If the media is inserted in a different orientation then
        any printing or stamping may be on the wrong side, upside down or both.
        This value is reported based on the customer's perspective."
    ::= { xfsIPMCapabilitiesEntry 22 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.23
xfsIPMCapabilitiesItemsTakenSensorInput OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the input position can detect when items at the
        exit position are taken by the user. This field relates to output and
        refused positions, so will always be set to FALSE for input positions."
    ::= { xfsIPMCapabilitiesEntry 23 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.24
xfsIPMCapabilitiesItemsInsertedSensorInput OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the input position has the ability to detect when
        items have been inserted by the user. This field relates to all input
        positions, so will always be set to FALSE for output and refused
        positions."
    ::= { xfsIPMCapabilitiesEntry 24 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.25
xfsIPMCapabilitiesRetractAreasInput OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION
    "It specifies the areas to which items may be retracted from the input
    position."
::= { xfsIPMCapabilitiesEntry 25 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.26
xfsIPMCapabilitiesItemsTakenSensorOutput OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the output position can detect when items at the
        exit position are taken by the user. This field relates to output and
        refused positions, so will always be set to FALSE for input positions."
    ::= { xfsIPMCapabilitiesEntry 26 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.27
xfsIPMCapabilitiesItemsInsertedSensorOutput OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the output position has the ability to detect when
        items have been inserted by the user. This field relates to all input
        positions, so will always be set to FALSE for output and refused
        positions."
    ::= { xfsIPMCapabilitiesEntry 27 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.28
xfsIPMCapabilitiesRetractAreasOutput OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the areas to which items may be retracted from the output
        position."
    ::= { xfsIPMCapabilitiesEntry 28 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.29
xfsIPMCapabilitiesItemsTakenSensorRefused OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the refused position can detect when items at the
        exit position are taken by the user. This field relates to output and
        refused positions, so will always be set to FALSE for input positions."
    ::= { xfsIPMCapabilitiesEntry 29 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.30
xfsIPMCapabilitiesItemsInsertedSensorRefused OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether the refused position has the ability to detect when
        items have been inserted by the user. This field relates to all input
        positions, so will always be set to FALSE for output and refused
        positions."
    ::= { xfsIPMCapabilitiesEntry 30 }

```



```

-- 1.3.6.1.4.1.16213.2.16.1.7.1.31
xfsIPMCapabilitiesRetractAreasRefused OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the areas to which items may be retracted from the refused
        position."
    ::= { xfsIPMCapabilitiesEntry 31 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.32
xfsIPMCapabilitiesGuidanceMediaIn OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the capability of the media input guidelight."
    ::= { xfsIPMCapabilitiesEntry 32 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.33
xfsIPMCapabilitiesGuidanceMediaOut OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the capability of the media output guidelight."
    ::= { xfsIPMCapabilitiesEntry 33 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.34
xfsIPMCapabilitiesGuidanceMediaRefused OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the capability of the media refused guidelight."
    ::= { xfsIPMCapabilitiesEntry 34 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.35
xfsIPMCapabilitiesPowerSaveControl OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies whether or not power saving control is available."
    ::= { xfsIPMCapabilitiesEntry 35 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.36
xfsIPMCapabilitiesImageAfterEndorse OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Specifies whether the device can generate an image after text is printed
        on the media item. If TRUE then the generation of the image can be
        specified using WFS_CMD_IPM_GET_IMAGE_AFTER_PRINT command. If FALSE, this
        functionality is not available. This capability applies to media items
        whose destination is a media bin; the fwReturnedItemsProcessing
        capability indicates whether this functionality is supported for media
        items that are to be returned to the customer."
    ::= { xfsIPMCapabilitiesEntry 36 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.37

```

```

xfsIPMCapabilitiesReturnedItemsProcessing OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It specifies the processing that this device supports for media items
        that are identified to be returned to the customer using the
        WFS_CMD_IPM_SET_DESTINATION command."
    ::= { xfsIPMCapabilitiesEntry 37 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.38
xfsIPMCapabilitiesMixedMode OBJECT-TYPE
    SYNTAX IxfsIPMCapabilitiesMixedMode
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This integer variable specifies whether or not the device supports
        accepting and processing items other than the types defined in the IPM
        specification."
    ::= { xfsIPMCapabilitiesEntry 38 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.39
xfsIPMCapabilitiesMixedDepositAndRollback OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This TruthValue variable specifies whether the device can deposit one
        type of media and rollback the other in the same Mixed Media
        transaction."
    ::= { xfsIPMCapabilitiesEntry 39 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.40
xfsIPMCapabilitiesAntiFraudModule OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This TruthValue variable specifies whether or not an anti-fraud module
        is available."
    ::= { xfsIPMCapabilitiesEntry 40 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.41
xfsIPMCapabilitiesSynchronizableCommands OBJECT-TYPE
    SYNTAX OCTET STRING
    ACCESS read-only
    STATUS current
    DESCRIPTION
        "This holds the list of execute command IDs that can be synchronized. It
        is an OCTET STRING with each command ID separated by a NULL terminator.
        The field itself is terminated by a double NULL terminator."
    ::= { xfsIPMCapabilitiesEntry 41 }

-- 1.3.6.1.4.1.16213.2.16.1.7.1.100
xfsIPMCapabilitiesExtraCapability OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "It contains the vendor dependent additional device capability
        information as an OCTET STRING. The information is returned as a series
        of 'key=value' strings. Each string is null-terminated, with the final
        string terminating with two null characters."
    ::= { xfsIPMCapabilitiesEntry 100 }

```

```

-- *****
-- IPM Sub Device Capabilities Table
--
-- The ASN.1 prefix for Version 1 of IPM is: 1.3.6.1.4.1.16213.2.16.1.8
-- *****
-- 1.3.6.1.4.1.16213.2.16.1.8
xfsIPMSubDeviceCapabilitiesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsIPMSubDeviceCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the IPM Sub DeviceCapabilities
        table."
    ::= { xfsIPMV1 8 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1
xfsIPMSubDeviceCapabilitiesEntry OBJECT-TYPE
    SYNTAX XfsIPMSubDeviceCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "IPM Bin Capabilities Table Entry."
    INDEX { xfsIPMSubDeviceCapabilitiesManagedServiceName ,
xfsIPMSubDeviceCapabilitiesNumberSubDevices, xfsIPMSubDeviceCapabilitiesIndex }
    ::= { xfsIPMSubDeviceCapabilitiesTable 1 }

XfsIPMSubDeviceCapabilitiesEntry ::=
    SEQUENCE {
        xfsIPMSubDeviceCapabilitiesManagedServiceName
            DisplayString,
        xfsIPMSubDeviceCapabilitiesIndex
            INTEGER,
        xfsIPMSubDeviceCapabilitiesMBBinNumber
            Integer32,
        xfsIPMSubDeviceCapabilitiesMBPositionName
            OCTET STRING,
        xfsIPMSubDeviceCapabilitiesMBHardwareSensors
            TruthValue,
        xfsIPMSubDeviceCapabilitiesMBItemSensors
            TruthValue,
        xfsIPMSubDeviceCapabilitiesMBExtra
            OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.1
xfsIPMSubDeviceCapabilitiesManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsIPMSubDeviceCapabilitiesEntry 1 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.2
xfsIPMSubDeviceCapabilitiesIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Index into the array of bins supported."
    ::= { xfsIPMSubDeviceCapabilitiesEntry 2 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.3
xfsIPMSubDeviceCapabilitiesMBBinNumber OBJECT-TYPE

```

```

SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Contains the media bin number. Each structure has a unique number
    starting with a value of one (1) for the first structure, and
    incrementing by one for each subsequent structure."
 ::= { xfsIPMSubDeviceCapabilitiesEntry 3 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.4
xfsIPMSubDeviceCapabilitiesMBPositionName OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains the physical bin position name where the bin is inserted."
    ::= { xfsIPMSubDeviceCapabilitiesEntry 4 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.5
xfsIPMSubDeviceCapabilitiesMBHardwareSensors OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "A capability that specifies whether or not the threshold event,
        WFS_USRE_IPM_MEDIABINTHRESHOLD (WFS_IPM_STATMBHIGH) can be generated
        based on hardware sensors in the device."
    ::= { xfsIPMSubDeviceCapabilitiesEntry 5 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.6
xfsIPMSubDeviceCapabilitiesMBItemSensors OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "A capability that specifies whether or not the threshold event,
        WFS_USRE_IPM_MEDIABININFOCHANGED (WFS_IPM_STATMBEMPTY) can be generated
        based on hardware sensors in the device."
    ::= { xfsIPMSubDeviceCapabilitiesEntry 6 }

-- 1.3.6.1.4.1.16213.2.16.1.8.1.100
xfsIPMSubDeviceCapabilitiesMBExtra OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Contains a list of vendor-specific, or any other extended, 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."
    ::= { xfsIPMSubDeviceCapabilitiesEntry 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
-- *****
-- 1.3.6.1.4.1.16213.3.0.116

```

```

xfsIPMDetailedDSCTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapEvent,
        xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsIPMStatusDevice,
xfsIPMStatusNumberSubDevices, xfsIPMStatusAcceptor,
        xfsIPMStatusMedia, xfsIPMStatusToner, xfsIPMStatusInk,
xfsIPMStatusFrontImageScanner, xfsIPMStatusBackImageScanner,
        xfsIPMStatusMICRReader, xfsIPMStatusStacker, xfsIPMStatusRebuncher,
xfsIPMStatusMediaFeeder, xfsIPMStatusShutterInput,
        xfsIPMStatusPositionInput, xfsIPMStatusTransportInput,
xfsIPMStatusTransportMediaInput, xfsIPMStatusShutterOutput,
xfsIPMStatusPositionOutput,
        xfsIPMStatusTransportOutput, xfsIPMStatusTransportMediaOutput,
xfsIPMStatusShutterRefused, xfsIPMStatusPositionRefused,
xfsIPMStatusTransportRefused,
        xfsIPMStatusTransportMediaRefused, xfsIPMStatusGuidanceMediaInput,
xfsIPMStatusGuidanceMediaOutput, xfsIPMStatusGuidanceMediaRefused,
xfsIPMStatusExtraStatus,
        xfsIPMStatusDevicePosition, xfsIPMStatusPowerSaveRecoveryTime,
xfsIPMStatusMixedMode, xfsIPMStatusAntiFraudModule, xfsIPMStatusJammedShutterInput,
xfsIPMStatusJammedShutterOutput,
        xfsIPMStatusJammedShutterRefused}
    STATUS current
    DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
    ::= { xfsTrapV2 116 }

-- 1.3.6.1.4.1.16213.3.0.216
xfsIPMSubDeviceTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapManagedServiceName, xfsCommonTrapManagedServiceClass,
xfsCommonTrapManagedServiceClassName, xfsCommonTrapManagedServiceType,
xfsCommonTrapManagedServiceOid,
        xfsCommonTrapPhysicalDeviceName, xfsCommonTrapDeviceVendor,
xfsCommonTrapMIBVersion, xfsCommonTrapEvent, xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsIPMSubDeviceIndex, xfsIPMSubDeviceMBBinNumber,
xfsIPMSubDeviceMBPositionName, xfsIPMSubDeviceMBType,
        xfsIPMSubDeviceMBMediaType, xfsIPMSubDeviceMBBinID,
xfsIPMSubDeviceMBMediaInCount, xfsIPMSubDeviceMBCount,
xfsIPMSubDeviceMBRetractOperations,
        xfsIPMSubDeviceMBHardwareSensor, xfsIPMSubDeviceMBMaximumItems,
xfsIPMSubDeviceMBMaximumRetractOperations, xfsIPMSubDeviceMBStatus,
xfsIPMSubDeviceMBExtraStatus, xfsCommonTrapSysName
    }
    STATUS current
    DESCRIPTION
        "This trap indicates a change in the status of sub-device within
        a managed service."
    ::= { xfsTrapV2 216 }

-- 1.3.6.1.4.1.16213.3.0.316
xfsIPMResetDeviceCompleteTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsIPMStatusDevice, xfsIPMStatusNumberSubDevices,
xfsIPMStatusAcceptor, xfsIPMStatusMedia,
        xfsIPMStatusToner, xfsIPMStatusInk, xfsIPMStatusFrontImageScanner,
xfsIPMStatusBackImageScanner, xfsIPMStatusMICRReader,
        xfsIPMStatusStacker, xfsIPMStatusRebuncher, xfsIPMStatusMediaFeeder,
xfsIPMStatusShutterInput, xfsIPMStatusPositionInput,

```

```

        xfsIPMStatusTransportInput, xfsIPMStatusTransportMediaInput,
xfsIPMStatusShutterOutput, xfsIPMStatusPositionOutput, xfsIPMStatusTransportOutput,
        xfsIPMStatusTransportMediaOutput, xfsIPMStatusShutterRefused,
xfsIPMStatusPositionRefused, xfsIPMStatusTransportRefused,
xfsIPMStatusTransportMediaRefused,
        xfsIPMStatusGuidanceMediaInput, xfsIPMStatusGuidanceMediaOutput,
xfsIPMStatusGuidanceMediaRefused, xfsIPMStatusExtraStatus,
xfsIPMStatusDevicePosition,
        xfsIPMStatusPowerSaveRecoveryTime, xfsIPMStatusMixedMode,
xfsIPMStatusAntiFraudModule, xfsIPMStatusJammedShutterInput,
xfsIPMStatusJammedShutterOutput,
        xfsIPMStatusJammedShutterRefused }
    STATUS current
    DESCRIPTION
        "This trap indicates the Reset action has complete and reports the
        state of the device after the reset."
    ::= { xfsTrapV2 316 }

END
--
--
--
-- SMIV2_xfsIPM.mib
--

```

5 Appendix B - C-Header files

The following paragraph contains the header file that can be used to implement the IPM SNMP support. Following the header file is the content of the header file.

5.1 XFSIPM.H

```

/*****
*
* xfsmibipm.h          CEN/XFS - MIB IPM
*
*          Version 3.30  --  February 6, 2017
*
*****/

#ifndef __inc_xfsmibipm_h
#define __inc_xfsmibipm_h

#ifdef __cplusplus
extern "C" {
#endif

/*****
    IPM Status #defines
*****/
enum IxfsIPMStatusAcceptor
{
    xfsIPMAccBinOK = 1,
    xfsIPMAccBinState,
    xfsIPMAccBinStop,
    xfsIPMAccBinUnknown
} xfsIPMStatusAcceptor;

enum IxfsIPMStatusMedia
{
    xfsIPMMediaPresent = 1,
    xfsIPMMediaNotPresent,
    xfsIPMMediaJammed,

```

```

        xfsIPMMediaNotSupported,
        xfsIPMMediaUnknown,
        xfsIPMMediaPosition
    } xfsIPMStatusMedia;

enum IxfsIPMStatusToner
{
    xfsIPMTonerFull = 1,
    xfsIPMTonerLow,
    xfsIPMTonerOut,
    xfsIPMTonerNotSupp,
    xfsIPMTonerUnknown
} xfsIPMStatusToner;

enum IxfsIPMStatusInk
{
    xfsIPMInkFull = 1,
    xfsIPMInkLow,
    xfsIPMInkOut,
    xfsIPMInkNotSupported,
    xfsIPMInkUnknown
} xfsIPMStatusInk;

enum IxfsIPMStatusScanner
{
    xfsIPMScannerOk = 1,
    xfsIPMScannerFading,
    xfsIPMScannerInop,
    xfsIPMScannerNotSupported,
    xfsIPMScannerUnknown
} xfsIPMStatusScanner;

enum IxfsIPMStatusMICRReader
{
    xfsIPMMICROk = 1,
    xfsIPMMICRFading,
    xfsIPMMICRInop,
    xfsIPMMICRNotSupported,
    xfsIPMMICRUnknown
} xfsIPMStatusMICRReader;

enum IxfsIPMStatusStacker
{
    xfsIPMStackerEmpty = 1,
    xfsIPMStackerNotEmpty,
    xfsIPMStackerFull,
    xfsIPMStackerInop,
    xfsIPMStackerUnknown,
    xfsIPMStackerNotSupported
} xfsIPMStatusStacker;

enum IxfsIPMStatusReBuncher
{
    xfsIPMRebuncherEmpty = 1,
    xfsIPMRebuncherNotEmpty,
    xfsIPMRebuncherFull,
    xfsIPMRebuncherInop,
    xfsIPMRebuncherUnknown,
    xfsIPMRebuncherNotSupported
} xfsIPMStatusReBuncher;

```

```
enum IxfsIPMStatusMediaFeeder
{
    xfsIPMMediaFeederEmpty = 1,
    xfsIPMMediaFeederNotEmpty,
    xfsIPMMediaFeederInop,
    xfsIPMMediaFeederUnknown,
    xfsIPMMediaFeederNotSupported
} xfsIPMStatusMediaFeeder;

enum IxfsIPMStatusShutter
{
    xfsIPMShutterClosed = 1,
    xfsIPMShutterOpen,
    xfsIPMShutterJammed,
    xfsIPMShutterUnknown,
    xfsIPMShutterNotSupported
} xfsIPMStatusShutter;

enum IxfsIPMStatusPosition
{
    xfsIPMPSEmpty = 1,
    xfsIPMPSEmptyNotSupported,
    xfsIPMPSEmptyUnknown,
    xfsIPMPSEmptyNotSupported
} xfsIPMStatusPosition;

enum IxfsIPMStatusTransport
{
    xfsIPMTPOK = 1,
    xfsIPMTPIInop,
    xfsIPMTPIUnknown,
    xfsIPMTPINotSupported
} xfsIPMStatusTransport;

enum IxfsIPMStatusTransportMedia
{
    xfsIPMTPMediaEmpty = 1,
    xfsIPMTPMediaNotEmpty,
    xfsIPMTPMediaUnknown,
    xfsIPMTPMediaNotSupported
} xfsIPMStatusTransportMedia;

enum IxfsIPMStatusDevicePosition
{
    xfsIPMDeviceInPosition = 1,
    xfsIPMDeviceNotInPosition,
    xfsIPMDevicePosUnknown,
    xfsIPMDevicePosNotSupported
} xfsIPMStatusDevicePosition;

enum IxfsIPMMixedModeStatus
{
    xfsIPMMixedMediaNotActive = 1,
    xfsIPMCIMMixedMedia
} xfsIPMMixedModeStatus;

enum IxfsIPMAntiFraudModuleStatus
{
    xfsIPMAFMNotSupported = 1,
    xfsIPMAFMOK,
    xfsIPMAFMInop,
    xfsIPMAFMDeviceDetected,
```



```

        xfsIPMAFMUnknown
    } xfsIPMAntiFraudModuleStatus;

enum IxfsIPMStatusJammedShutter
{
    xfsIPMShutterPositionNotSupported = 1,
    xfsIPMShutterPositionNotJammed,
    xfsIPMShutterPositionOpen,
    xfsIPMShutterPositionPartiallyOpen,
    xfsIPMShutterPositionClosed,
    xfsIPMShutterPositionUnknown
} xfsIPMStatusJammedShutter;

/*****
    IPM SubDevice #defines
*****/

enum IxfsIPMSubDeviceMBMediaType
{
    xfsIPMMediaTypeIPM = 2,
    xfsIPMMediaTypeCompound

} xfsIPMSubDeviceMBMediaType;

enum IxfsIPMSubDeviceMBStatus
{
    xfsIPMMBStatOK = 1,
    xfsIPMMBStatFull,
    xfsIPMMBStatHigh,
    xfsIPMMBStatInop,
    xfsIPMMBStatMissing,
    xfsIPMMBStatUnknown,
    xfsIPMMBStatEmpty

} xfsIPMSubDeviceMBStatus;

/*****
    IPM Capabilities #defines
*****/

enum IxfsIPMCapabilitiesDeviceType
{
    xfsIPMSingleMediaInput = 2,
    xfsIPMBunchedMediaInput

} xfsIPMCapabilitiesDeviceType;

enum IxfsIPMCapabilitiesDefaultScanColor
{
    xfsIPMScanColorRed      = 2,
    xfsIPMScanColorBlue     = 3,
    xfsIPMScanColorGreen    = 5,
    xfsIPMScanColorYellow   = 9,
    xfsIPMScanColorWhite    = 17

} xfsIPMCapabilitiesDefaultScanColor;

enum IxfsIPMCapabilitiesMixedMode
{
    xfsIPMCapMixedMediaNotSupp    = 1,
    xfsIPMCapCIMMixedMedia

} xfsIPMItemInfoTypes;

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

```

```

*****/
#define xfsIPMStatusManagedServiceName (1)
#define xfsIPMStatusNumberSubDevices (2)
#define xfsIPMStatusDevice (3)
#define xfsIPMStatusAcceptor (4)
#define xfsIPMStatusMedia (5)
#define xfsIPMStatusToner (6)
#define xfsIPMStatusInk (7)
#define xfsIPMStatusFrontImageScanner (8)
#define xfsIPMStatusBackImageScanner (9)
#define xfsIPMStatusMICRReader (10)
#define xfsIPMStatusStacker (11)
#define xfsIPMStatusReBuncher (12)
#define xfsIPMStatusMediaFeeder (13)
#define xfsIPMStatusShutterInput (14)
#define xfsIPMStatusPositionInput (15)
#define xfsIPMStatusTransportInput (16)
#define xfsIPMStatusTransportMediaInput (17)
#define xfsIPMStatusShutterOutput (18)
#define xfsIPMStatusPositionOutput (19)
#define xfsIPMStatusTransportOutput (20)
#define xfsIPMStatusTransportMediaOutput (21)
#define xfsIPMStatusShutterRefused (22)
#define xfsIPMStatusPositionRefused (23)
#define xfsIPMStatusTransportRefused (24)
#define xfsIPMStatusTransportMediaRefused (25)
#define xfsIPMStatusGuidanceMediaInput (26)
#define xfsIPMStatusGuidanceMediaOutput (27)
#define xfsIPMStatusGuidanceMediaRefused (28)
#define xfsIPMStatusDevicePosition (29)
#define xfsIPMStatusPowerSaveRecoveryTime (30)
#define xfsIPMStatusMixedMode (31)
#define xfsIPMStatusAntiFraudModule (32)
#define xfsIPMStatusJammedShutterInput (33)
#define xfsIPMStatusJammedShutterOutput (34)
#define xfsIPMStatusJammedShutterRefused (35)
#define xfsIPMStatusExtraStatus (100)

/*****
*
* MIB Variables for the SubDevice Table
*
*****/
#define xfsIPMSubDeviceManagedServiceName (1)
#define xfsIPMSubDeviceIndex (2)
#define xfsIPMSubDeviceMBNumber (3)
#define xfsIPMSubDeviceMBPositionName (4)
#define xfsIPMSubDeviceMBType (5)
#define xfsIPMSubDeviceMBMediaType (6)
#define xfsIPMSubDeviceMBID (7)
#define xfsIPMSubDeviceMBMediaInCount (8)
#define xfsIPMSubDeviceMBCount (9)
#define xfsIPMSubDeviceMBRetractOperations (10)
#define xfsIPMSubDeviceMBHardwareSensor (11)
#define xfsIPMSubDeviceMBMaximumItems (12)
#define xfsIPMSubDeviceMBMaximumRetractOperations (13)
#define xfsIPMSubDeviceMBStatus (14)
#define xfsIPMSubDeviceMBExtraStatus (100)

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

/*****
*

```

```

*          MIB Variables for the Capabilities Table          *
*
*****/
#define xfsIPMCapabilitiesManagedServiceName (1)
#define xfsIPMCapabilitiesDeviceType (2)
#define xfsIPMCapabilitiesCompoundDevice (3)
#define xfsIPMCapabilitiesMaxMediaOnStacker (4)
#define xfsIPMCapabilitiesPrintSize (5)
#define xfsIPMCapabilitiesStamping (6)
#define xfsIPMCapabilitiesRescan (7)
#define xfsIPMCapabilitiesPresentControl (8)
#define xfsIPMCapabilitiesApplicationRefuse (9)
#define xfsIPMCapabilitiesRetractLocations (10)
#define xfsIPMCapabilitiesResetControl (11)
#define xfsIPMCapabilitiesRetractCountsItems (12)
#define xfsIPMCapabilitiesImageType (13)
#define xfsIPMCapabilitiesFrontImageColorFormat (14)
#define xfsIPMCapabilitiesBackImageColorFormat (15)
#define xfsIPMCapabilitiesFrontScanColor (16)
#define xfsIPMCapabilitiesDefaultFrontScanColor (17)
#define xfsIPMCapabilitiesBackScanColor (18)
#define xfsIPMCapabilitiesDefaultBackScanColor (19)
#define xfsIPMCapabilitiesCodelineFormat (20)
#define xfsIPMCapabilitiesDataSource (21)
#define xfsIPMCapabilitiesOrientation (22)
#define xfsIPMCapabilitiesItemSensorInput (23)
#define xfsIPMCapabilitiesItemInsertedSensorInput (24)
#define xfsIPMCapabilitiesInputRetractAreas (25)
#define xfsIPMCapabilitiesItemSensorOutput (26)
#define xfsIPMCapabilitiesItemInsertedSensorOutput (27)
#define xfsIPMCapabilitiesOutputRetractAreas (28)
#define xfsIPMCapabilitiesItemSensorRefused (29)
#define xfsIPMCapabilitiesItemInsertedSensorRefused (30)
#define xfsIPMCapabilitiesRefusedRetractAreas (31)
#define xfsIPMCapabilitiesMediaInGuideLights (32)
#define xfsIPMCapabilitiesMediaOutGuideLights (33)
#define xfsIPMCapabilitiesMediaRefusedGuideLights (34)
#define xfsIPMCapabilitiesPowerSaveControl (35)
#define xfsIPMCapabilitiesImageAfterEndorse (36)
#define xfsIPMCapabilitiesReturnedItemsProcessing (37)
#define xfsIPMCapabilitiesMixedMode (38)
#define xfsIPMCapabilitiesMixedDepositAndRollback (39)
#define xfsIPMCapabilitiesAntiFraudModule (40)
#define xfsIPMCapabilitiesSynchronizableCommands (41)
#define xfsIPMCapabilitiesExtraCapability (100)

/*****
*
*          MIB Variables for the Bin Capabilities Table          *
*
*****/
#define xfsIPMBinCapabilitiesManagedServiceName (1)
#define xfsIPMBinCapabilitiesIndex (2)
#define xfsIPMBinCapabilitiesMBBinNumber (3)
#define xfsIPMBinCapabilitiesMBPositionName (4)
#define xfsIPMBinCapabilitiesMBHardwareSensors (5)
#define xfsIPMBinCapabilitiesMBItemSensors (6)
#define xfsIPMBinCapabilitiesMBExtra (100)

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

```