CEN

WORKSHOP

AGREEMENT

CWA 16926-34

February 2017

ICS 35.240.40

English version

Extensions for Financial Services (XFS) interface specification – Release 3.30 – Part 34: XFS MIB Device Specific Definitions – Check Reader/Scanner Device Class MIB 3.30

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

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

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

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

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

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

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

© 2017 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.
Ref. No. CWA 16926-34:2017 D/E/F

Table of Contents

Foreword **3**

1. Introduction **7**

2. XFS CHK MIB variables **10**

2.1 XFS CHK STATUS TABLE 10
2.1.1 xfsCHKStatusTable: States *10*
2.2 XFS CHK SUB DEVICE TABLE 12
2.3 XFS CHK ERROR TABLE 12
2.4 XFS CHK RESET TABLE 13
2.5 XFS CHK RESET DEVICE TABLE 13
2.6 XFS CHK CAPABILITIES TABLE 14
2.6.1 xfsCHKCapabilitiesTable: Capabilities *15*

3. CHK Traps **18**

3.1 CHK DETAILED DEVICE STATUS CHANGE TRAP 18
3.1.1 CHK Detailed Device Status Change Trap Format *18*
3.1.2 CHK Detailed Device Status Change Trap: an example *20*
3.2 CHK SUB-DEVICE STATUS CHANGE TRAP 21
3.3 CHK RESET DEVICE COMPLETE TRAP 21
3.3.1 CHK Reset Device Complete Trap Format *21*
3.3.2 CHK Reset Device Complete: an example *23*

4. Appendix A - CHK MIB sub-tree **25**

4.1 CHK MIB IN SMIV2 AND SMIV1 ASN-1 FORMAT 25

5. Appendix B - C-Header files **38**

5.1 XFSMIBCHK.H 38

Foreword

This CWA is revision 3.30 of the XFS interface specification.

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

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

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

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

Part 2: Service Classes Definition - Programmer's Reference

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Parts 19 - 28: Reserved for future use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Part 42: Reserved for future use.

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

Part 44: XFS MIB Application Management

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

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

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

Parts 48 - 60 are reserved for future use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is

available online from http://www.cen.eu/work/areas/ict/ebusiness/pages/ws-xfs.aspx.

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

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

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

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

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

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

Revision History:

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

1. Introduction

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

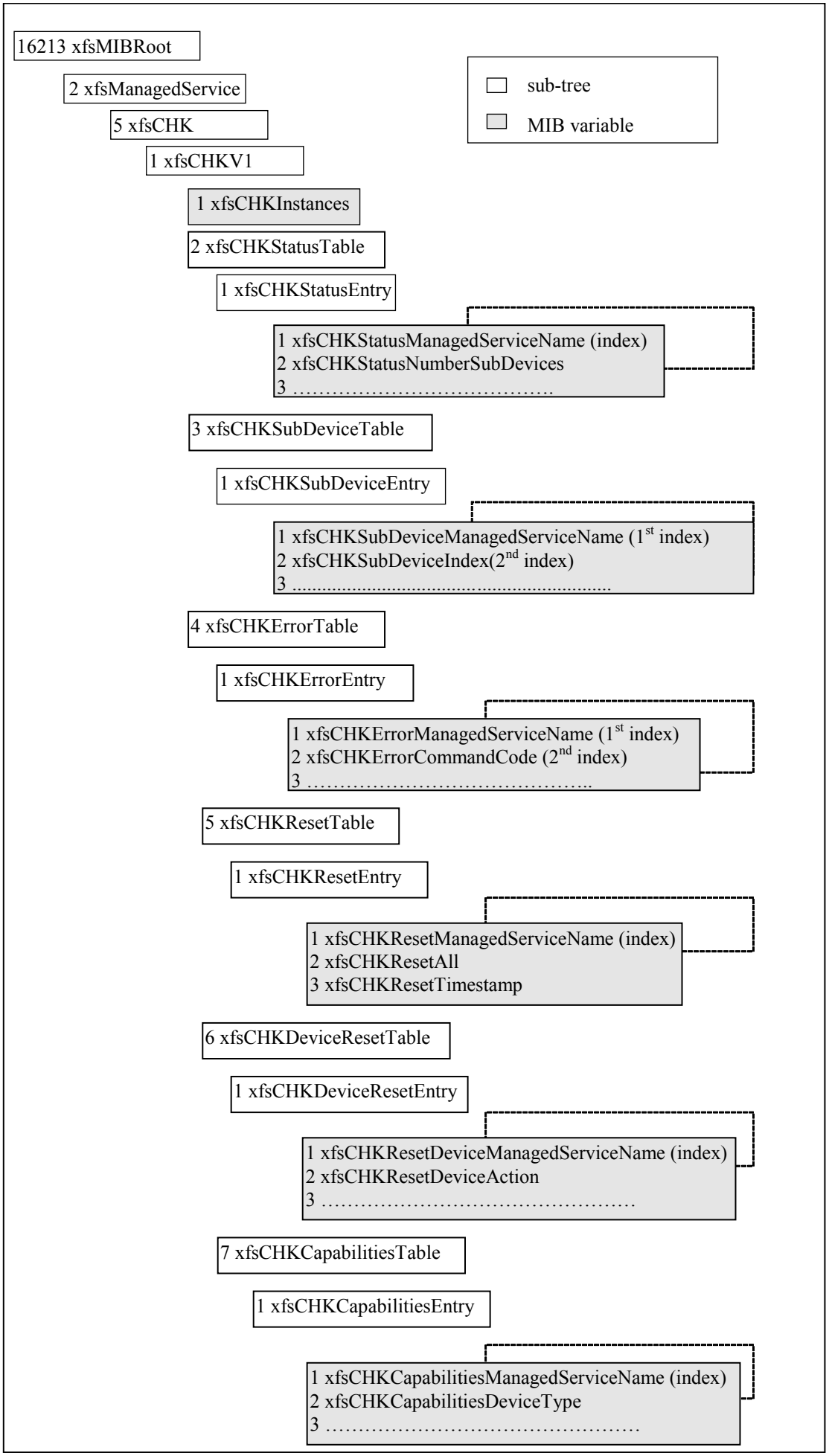
xfsMIBRoot

- xfsManagedService (2)
  - xfsCHK (5)
    - xfsCHKV1 (1)

The xfsCHKV1 sub-tree contains the following variables:

* *xfsCHKInstances(1)* is the number of managed services for the CHK class installed on the XFS subsystem. It is a 32 bit numerical field.
* *xfsCHKStatusTable(2)* identifies the table for the CHK variables.
* *xfsCHKSubDeviceTable(3)* not applicable to the CHK device.
* *xfsCHKErrorTable(4)* identifies the table for the CHK error counters.
* *xfsCHKResetTable(5)* identifies the table for the CHK reset variable.
* *xfsCHKResetDeviceTable(6)* identifies the table for the CHK reset device variables.
* *xfsCHKCapabilitiesTable(7)* identifies the table for the CHK 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 *xfsCHKV1* sub-tree.


16213 xfsMIBRoot
2 xfsManagedService
5 xfsCHK
1 xfsCHKV1
1 xfsCHKInstances
2 xfsCHKStatusTable
1 xfsCHKStatusEntry
1 xfsCHKStatusManagedServiceName (index)
2 xfsCHKStatusNumberSubDevices
3 ……
3 xfsCHKSubDeviceTable
1 xfsCHKSubDeviceEntry
1 xfsCHKSubDeviceManagedServiceName (1st index)
2 xfsCHKSubDeviceIndex(2nd index)
3 .....
4 xfsCHKErrorTable
1 xfsCHKErrorEntry
1 xfsCHKErrorManagedServiceName (1st index)
2 xfsCHKErrorCommandCode (2nd index)
3 ……..
5 xfsCHKResetTable
1 xfsCHKResetEntry
1 xfsCHKResetManagedServiceName (index)
2 xfsCHKResetAll
3 xfsCHKResetTimestamp
6 xfsCHKDeviceResetTable
1 xfsCHKDeviceResetEntry
1 xfsCHKResetDeviceManagedServiceName (index)
2 xfsCHKResetDeviceAction
3 ……
7 xfsCHKCapabilitiesTable
1 xfsCHKCapabilitiesEntry
1 xfsCHKCapabilitiesManagedServiceName (index)
2 xfsCHKCapabilitiesDeviceType
3 ……
sub-tree
MIB variable

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

2. XFS CHK MIB variables

This section describes the MIB variables for the tables of the CHK Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Check Reader/Scanner 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 the all response counters to be reset.
- Each managed service has a Reset Device table that allows the WFS_CMD_CHK_RESET command to be executed from the management station.

2.1 XFS CHK Status Table

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

xfsCHKStatusManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the CHK class.

As an example, the identifier for the device status value of *xfsCHKStatusMedia(4)* for a device with managed service name equal to "CheckReader1" is as follows:

Character	C	h	E	c	k	R	e	a	d	e	r	1
ASCII Hex	43	68	65	63	6B	52	65	61	64	65	72	31
ASCII Dec	67	104	101	99	107	82	101	97	100	101	114	49

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

xfsMIBRoot.2.5.1.2.1.4.12.67.104.101.99.107.82.101.97.100.101.114.49

2.1.1 xfsCHKStatusTable: States

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

xfsCHKStatusManagedServiceName (1)
Uniquely identifies the managed service.

xfsCHKStatusNumberSubDevices (2)
Defines how many sub-devices the service has. This is always 0 (zero) in the CHK device class.

xfsCHKStatusDevice (3)
It contains the device state. It is a numeric type field. Allowed values are as follows:

Value	Meaning
xfsDevOnline (1)	The device is present, powered on and online (i.e., operational, not busy processing a request and not in an error state).
xfsDevOffline (2)	The device is offline (e.g., the operator has taken the device offline by turning a switch).
xfsDevPowerOff (3)	The device is powered off or physically not connected.
xfsDevNoDevice (4)	There is no device intended to be there; e.g. this type of self service machine does not contain such a device or it is internally not configured.
xfsDevHWError (5)	The device is inoperable due to a hardware error.
xfsDevUserError (6)	The device is inoperable because 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 has detected a potential fraud attempt and is capable of remaining in service.

xfsCHKStatusMedia (4)

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

Value	Meaning
xfsCHKMediaNotSupp (1)	The capability to report the state of the check media is not supported by the device.
xfsCHKMediaNotPresent (2)	No media is inserted in device.
xfsCHKMediaRequired (3)	Insertion of media required.
xfsCHKMediaPresent (4)	Media inserted in device.
xfsCHKMediaJammed (5)	Media jam in device.
xfsCHKMediaEjected (6)	Media ejected.
xfsCHKMediaRetained (7)	Media retained.

xfsCHKStatusInk (5)

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

Value	Meaning
xfsCHKInkNotSupp (1)	Capability not supported by the device.
xfsCHKInkFull (2)	Ink supply in device is full.
xfsCHKInkLow (3)	Ink supply in device is low.
xfsCHKInkOut (4)	Ink supply in device is empty.

xfsCHKStatusGuidanceCheckUnit (6)

It contains the state of the guidance light on the check processing unit.

Value	XFS Name	Meaning
0x00000000	WFS_CHK_GUIDANCE_NOT_AVAILABLE	The status is not available.
0x00000001	WFS_CHK_GUIDANCE_OFF	The light is turned off.
0x00000004	WFS_CHK_GUIDANCE_SLOW_FLASH	The light is blinking slowly.
0x00000008	WFS_CHK_GUIDANCE_MEDIUM_FLASH	The light is blinking medium frequency.
0x00000010	WFS_CHK_GUIDANCE_QUICK_FLASH	The light is blinking quickly.
0x00000080	WFS_CHK_GUIDANCE_CONTINUOUS	The light is turned on continuous (steady).
0x00000100	WFS_CHK_GUIDANCE_RED	The light is red.
0x00000200	WFS_CHK_GUIDANCE_GREEN	The light is green.
0x00000400	WFS_CHK_GUIDANCE_YELLOW	The light is yellow.
0x00000800	WFS_CHK_GUIDANCE_BLUE	The light is blue.
0x00001000	WFS_CHK_GUIDANCE_CYAN	The light is cyan.
0x00002000	WFS_CHK_GUIDANCE_MAGENTA	The light is magenta.
0x00004000	WFS_CHK_GUIDANCE_WHITE	The light is white.
0x00100000	WFS_CHK_GUIDANCE_ENTRY	The light is in the entry state.

0x00200000	WFS_CHK_GUIDANCE_EXIT	The light is in the exit state.

xfsCHKStatusDevicePosition (7)

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

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

xfsCHKStatusPowerSaveRecoveryTime (8)

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.

xfsCHKStatusAntiFraudModule (9)

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

Value	Meaning
xfsCHKAFMNotSupported (1)	No anti-fraud module is available.
xfsCHKAFMOK (2)	Anti-fraud module is in a good state and no foreign device is detected.
xfsCHKAFMInop (3)	Anti-fraud module is inoperable.
xfsCHKAFMDeviceDetected (4)	Anti-fraud module detected the presence of a foreign device.
xfsCHKAFMUnknown (5)	The state of the anti-fraud module cannot be determined.

xfsCHKStatusExtraStatus (100)

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

2.2 XFS CHK Sub Device Table

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

2.3 XFS CHK Error Table

The *xfsCHKErrorTable(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:

xfsCHKErrorManagedServiceName
xfsCHKErrorCommandCode
xfsCHKErrorResponseCode

The *xfsCHKErrorTable* is defined as:

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

xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName in the general table, e.g. "CheckReader1".

- *xfsCHKErrorCommandCode(2)* is an index which identifies the command code that that response code is related to, e.g. WFS_CMD_CHK_PROCESS_FORM (501). It is a 32 bit numerical field.
- *xfsCHKErrorResponseCode(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_CHK_FORMNOTFOUND (-500) is represented by 500. It is a 32 bit numerical field.
- *xfsCHKErrorCount(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_CHK_FORMNOTFOUND (-500) error returned from the WFS_CMD_CHK_PROCESS_FORM (501) command for a device with managed service name equal to "CheckReader1" is as follows:

xfsMIBRoot.2.5.1.4.1.4.12.67.104.101.99.107.82.101.97.100.101.114.49.501.500

2.4 XFS CHK Reset Table

The *xfsCHKResetTable(5)* contains the *xfsCHKResetAll* and *xfsCHKResetTimestamp* variables and is indexed by the single variable, *xfsCHKResetManagedServiceName*. When the *xfsCHKResetAll* 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 *xfsCHKResetTable* is defined as:

- *xfsCHKResetManagedServiceName(1)* which provides the index to the service in question. It is Display String field. The *xfsCHKResetManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table, e.g. "CheckReader1".
- *xfsCHKResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsCHKResetAll* 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 *xfsCHKResetAll* variable will return 0 (zero).
- *xfsCHKResetTimestamp(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 Coordinated 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 "CheckReader1" by setting the value zero in the *xfsCHKResetAll* variable represented by:

xfsMIBRoot.2.5.1.5.1.2.12.67.104.101.99.107.82.101.97.100.101.114.49

2.5 XFS CHK Reset Device Table

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

The *xfsCHKResetDeviceAction* 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_CHK_RESET command will be issued.
4. Exclusive access to the device will be relinquished when the WFS_CMD_CHK_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 *xfsCHKResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request.

The *xfsCHKResetDeviceTable* is defined as:

- *xfsCHKResetDeviceManagedServiceName(1)* which provides the index to the service in question. It is a Display String field. The *xfsCHKResetDeviceManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table, e.g. "CheckReader1".
- *xfsCHKResetDeviceAction(2)* is a read-write variable. Issue of a Set command on the *xfsCHKResetDeviceAction* variable with value *executeReset(1)* will result in the device being reset as described above.
- *xfsCHKResetDeviceMediaControl(3)* is a read-only variable. This variable reports how any media found within the device is handled. The value of the *xfsCHKResetDeviceMediaControl* 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.
- *xfsCHKResetDeviceStatus(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 "CheckReader1" is reset by setting the *xfsCHKResetDeviceAction* variable represented by:

xfsMIBRoot.2.5.1.6.1.2.12.67.104.101.99.107.82.101.97.100.101.114.49

2.6 XFS CHK Capabilities Table

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

xfsCHKCapabilitiesManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the CHK class.

As an example, the identifier for the device status value of *xfsCHKCapabilitiesDeviceType(2)* for a device with managed service name equal to "CheckReader1" is as follows:

Character	C	h	e	c	k	R	e	a	d	e	r	1
ASCII Hex	43	68	65	63	6B	52	65	61	64	65	72	31
ASCII Dec	67	104	101	99	107	82	101	97	100	101	114	49

The OID of the above example is:

xfsMIBRoot.2.5.1.7.1.2.12.67.104.101.99.107.82.101.97.100.101.114.49

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

2.6.1 xfsCHKCapabilitiesTable: Capabilities

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

xfsCHKCapabilitiesManagedServiceName (1)
Uniquely identifies the managed service.

xfsCHKCapabilitiesDeviceType (2)
Specifies the type of the physical device; only current value as a integer is:

Value	Meaning
xfsCHKTypeChk (2)	Check Reader system.

xfsCHKCapabilitiesCompoundDevice (3)
Specifies if the logical device is part of a compound device in a TruthValue variable as follows:

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

xfsCHKCapabilitiesMICR (4)
Specifies if the logical device can read MICR on check in a TruthValue variable as follows:

Value	Meaning
True (1)	The device can read MICR on check.
False (2)	The device cannot read MICR on check.

xfsCHKCapabilitiesOCR (5)
Specifies if the logical device can read OCR on check in a TruthValue variable as follows:

Value	Meaning
True (1)	The device can read OCR on check.
False (2)	The device cannot read OCR on check.

xfsCHKCapabilitiesAutofeed (6)
Specifies whether device has autofeed or manual feed capability in a TruthValue variable as follows:

Value	Meaning
True (1)	The device has autofeed capability.
False (2)	The device has manual feed capability.

xfsCHKCapabilitiesEndorser (7)
Specifies whether programmer endorser present or absent in a TruthValue variable as follows:

Value	Meaning
True (1)	Programmer endorser present.
False(2)	Programmer endorser absent.

xfsCHKCapabilitiesEncoder (8)
Specifies whether encoder present or absent in a TruthValue variable as follows:

Value	Meaning
True (1)	Encoder present.
False (2)	Encoder absent.

xfsCHKCapabilitiesStamp (9)
Specifies the physical dimensions of the check where the endorser stamp can be used. A single value can be returned as an integer.

Value	Meaning
xfsCHKStampNone (2)	Device cannot stamp/endorse check.
xfsCHKStampFront (3)	Device can stamp/endorse front of check.
xfsCHKStampBack (4)	Device can stamp/endorse back of check.
xfsCHKStampBoth (5)	Device can stamp/endorse both sides of the check.

xfsCHKCapabilitiesImageCapture (10)

Specifies the physical dimensions that can be image captured. A single value can be returned as an integer.

Value	Meaning
xfsCHKICapNone (2)	Device cannot capture image.
xfsCHKICapFront (3)	Device can image capture front of check.
xfsCHKICapBack (4)	Device can image capture back of check.
xfsCHKICapBoth (5)	Device can image capture both sides of the check.

xfsCHKCapabilitiesFontNames (11)

Specifies the names of the fonts supported for reading; each is terminated with a null and the string is terminated with two nulls. Reserved font names include CMC7 and E13B.

xfsCHKCapabilitiesEncodeNames (12)

Specifies the names of the fonts supported for encoding; each is terminated with a null and the string is terminated with two nulls.

xfsCHKCapabilitiesCharSupport (13)

Specifies the character sets, in addition to single byte ASCII, that are supported by the Service Provider. This field will be set to a combination of hex values according to the values in the following table:

Value	XFS Name	Meaning
0x0001	WFS_CHK_ASCII	ASCII is supported for XFS forms.
0x0002	WFS_CHK_UNICODE	UNICODE is supported for XFS forms.

xfsCHKCapabilitiesGuidanceCheckUnit (14)

Specifies the state of the check unit guidance light. This field 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
0x00000000	WFS_CHK_GUIDANCE_NOT_AVAILABLE	There is no guidance light control available at this position.
0x00000001	WFS_CHK_GUIDANCE_OFF	The light can be off.
0x00000004	WFS_CHK_GUIDANCE_SLOW_FLASH	The light can blink slowly.
0x00000008	WFS_CHK_GUIDANCE_MEDIUM_FLASH	The light can blink medium frequency.
0x00000010	WFS_CHK_GUIDANCE_QUICK_FLASH	The light can blink quickly.
0x00000080	WFS_CHK_GUIDANCE_CONTINUOUS	The light can be continuous (steady).
0x00000100	WFS_CHK_GUIDANCE_RED	The light can be red.
0x00000200	WFS_CHK_GUIDANCE_GREEN	The light can be green.
0x00000400	WFS_CHK_GUIDANCE_YELLOW	The light can be yellow.
0x00000800	WFS_CHK_GUIDANCE_BLUE	The light can be blue.
0x00001000	WFS_CHK_GUIDANCE_CYAN	The light can be cyan.
0x00002000	WFS_CHK_GUIDANCE_MAGENTA	The light can be magenta.
0x00004000	WFS_CHK_GUIDANCE_WHITE	The light can be white.
0x00100000	WFS_CHK_GUIDANCE_ENTRY	The light can be in the entry state.
0x00200000	WFS_CHK_GUIDANCE_EXIT	The light can be in the exit state.

xfsCHKCapabilitiesPowerSaveControl (15)

Specifies the capability of the power saving control in a TruthValue variable as follows:

Value	Meaning
True (1)	PowerSaveControl available.
False (2)	PowerSaveControl not available.

xfsCHKCapabilitiesAntiFraudModule (16)

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

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

xfsCHKCapabilitiesSynchronizableCommands (17)

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

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

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

<null><null>

Possible values for each command ID are:

Value	XFS Name	Meaning
501	WFS_CMD_CHK_PROCESS_FORM	The WFS_CMD_CHK_PROCESS_FORM command can be synchronized.
502	WFS_CMD_CHK_RESET	The WFS_CMD_CHK_RESET command can be synchronized.
503	WFS_CMD_CHK_SET_GUIDANCE_LIGHT	The WFS_CMD_CHK_SET_GUIDANCE_LIGHT command can be synchronized.
504	WFS_CMD_CHK_POWER_SAVE_CONTROL	The WFS_CMD_CHK_POWER_SAVE_CONTROL command can be synchronized.

xfsCHKCapabilitiesExtraCapability (100)

Pointer to a list of vendor-specific, or any other extended, information. The information is returned as a series of *"key=value"* strings so that it is easily extensible by Service Providers. Each string is null-terminated, with the final string terminating with two null characters.

3. CHK Traps

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

3.1 CHK 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 CHK 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 CHK reflect the CHK Status Table as defined in section 2.1.

The detailed device status change event is only generated when the top level status changes within a managed service, i.e. the trap is generated when the *fwDevice* value in the WFS_INF_CHK_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 105 defines the trap as a CHK 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 CHK Detailed Device Status Change Trap Format

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

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName (1)
This variable binding contains the system generating the alarm, it is a Display String field. It corresponds to *lpszWorkstationName* in the device status change event data from the Service Provider.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName (2)
This variable binding represents the managed service name generating the alarm, it is a Display String field. The agent derives this field from the device status change event.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass (3)
This variable binding represents the XFS service class identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<Managed*ServiceName*>\class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CDM=3, etc. See the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document for a complete list of these values.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName (4)
This variable binding represents the XFS service class name generating the alarm, it is a Display String field. It corresponds to the three character representation of the XFS device class name, and it is useful for human interpretation of a trap. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\class.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType (5)
This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS_INF_CHK_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 CHK MIB class is represented by .1.3.6.1.4.1.16213.2.5

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.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevice**.xfsCHKStatusManagedServiceName (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.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusNumberSubDevices.**xfsCHKStatusManagedServiceName (14)
Defines how many sub-devices the service has. This is the number of retract bins the device supports.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusMedia.**xfsCHKStatusManagedServiceName (15)
It contains the media state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusInk.**xfsCHKStatusManagedServiceName (16)
It contains the ink state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusExtraStatus.**xfsCHKStatusManagedServiceName (17)
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.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusGuidanceCheckUnit.**xfsCHKStatusManagedServiceName (18)
It contains the state of the guidance light indicator on the check processing unit. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevicePosition.**xfsCHKStatusManagedServiceName (19)
It contains the device position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusPowerSaveRecoveryTime.**xfsCHKStatusManagedServiceName (20)
It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. It is a numeric type field. This value is zero if either the power saving mode has not been activated or no power save control is supported.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusAntiFraudModule.**xfsCHKStatusManagedServiceName (21)
It contains the state of the anti-fraud module. It is a numeric type field.

3.1.2 CHK Detailed Device Status Change Trap: an example

As an example, the following variable binding list represents a detailed device status change trap (6, 105) that is generated for a CHK with a managed service name of “CheckReader1”. It reports that the device is OFFLINE because the media is jammed.

xfsMIBRoot.3.1.3.1	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName)
	“SST System 1”
xfsMIBRoot.3.1.3.2	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	“CheckReader1”
xfsMIBRoot.3.1.3.3	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	5 (WFS_SERVICE_CLASS_CHK)
xfsMIBRoot.3.1.3.4	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	“CHK”
xfsMIBRoot.3.1.3.5	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	1 (WFS_CHK_TYPECHK)
xfsMIBRoot.3.1.3.6	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	“.1.3.6.1.4.1.16213.2.5”
xfsMIBRoot.3.1.3.7	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	“ABC Corp Check Reader”
xfsMIBRoot.3.1.3.8	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	“Best Devices Incorporated”
xfsMIBRoot.3.1.3.9	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	“1.10”
xfsMIBRoot.3.1.3.10	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent)
	4 (WFS_SYSE_DEVICE_STATUS)
xfsMIBRoot.3.1.3.11	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	“20/03/2003 15:40:53 -300”
xfsMIBRoot.3.1.3.12	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	“1.23”

xfsMIBRoot.2.5.1.2.1.**3**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevice**.xfsCHKStatusManagedServiceName)
	2 (WFS_STAT_DEVOFFLINE)
xfsMIBRoot.2.5.1.2.1.**2**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusNumberSubDevices**.xfsCHKStatusManagedServiceName)
	0 (No sub device)
xfsMIBRoot.2.5.1.2.1.**4**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusMedia**.xfsCHKStatusManagedServiceName)
	5 (xfsCHKMediaJammed)
xfsMIBRoot.2.5.1.2.1.**5**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusInk**.xfsCHKStatusManagedServiceName)
	3 (xfsCHKInkLow)
xfsMIBRoot.2.5.1.2.1.**100**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusExtraStatus**.xfsCHKStatusManagedServiceName)
	'\0"\0' (No extra data)
xfsMIBRoot.2.5.1.2.1.**6**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusGuidanceCheckUnit**.xfsCHKStatusManagedServiceName)
	1 (value corresponding to WFS_CHK_GUIDANCE_OFF)
xfsMIBRoot.2.5.1.2.1.7.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevicePosition**.xfsCHKStatusManagedServiceName)
	1 (xfsCHKDeviceInPosition)
xfsMIBRoot.2.5.1.2.1.**8**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusPowerSaveRecoveryTime**.xfsCHKStatusManagedServiceName)
	3 (3 seconds to recover from power saving mode)
xfsMIBRoot.2.5.1.2.1.**9**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusAntiFraudModule**.xfsCHKStatusManagedServiceName)
	2 (xfsCHKAFMOK)

3.2 CHK Sub-Device Status Change Trap

The CHK does not currently support any sub-devices so the CHK Sub-Device Status Change Trap is not currently defined. The SNMP Specific trap value 205 is reserved in case a sub-device is ever added to the CHK device class.

3.3 CHK Reset Device Complete Trap

On the CHK 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 305 defines the trap as a CHK Reset Device Complete trap.

3.3.1 CHK Reset Device Complete Trap Format

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

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

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

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass (3)
This variable binding represents the XFS service class identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<Managed*ServiceName*>\class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CDM=3, etc. See the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document for a complete list of these values.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName (4)
This variable binding represents the XFS service class name generating the alarm, it is a Display String field. It corresponds to the three character representation of the XFS device class name, and it is useful for human interpretation of a trap. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\class.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType (5)
This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS_INF_CHK_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 CHK MIB class is represented by .1.3.6.1.4.1.16213.2.5

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.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevice**.xfsCHKStatusManagedServiceName (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.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusNumberSubDevices.**xfsCHKStatusManagedServiceName (13)
Defines how many sub-devices the service has. This is the number of retract bins the device supports.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusMedia.**xfsCHKStatusManagedServiceName (14)
It contains the media state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusInk.**xfsCHKStatusManagedServiceName (15)
It contains the ink state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusExtraStatus.**xfsCHKStatusManagedServiceName (16)
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.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusGuidanceCheckUnit.**xfsCHKStatusManagedServiceName (17)
It contains the state of the guidance light indicator on the check processing unit. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.
xfsCHKStatusDevicePosition.xfsCHKStatusManagedServiceName (18)
It contains the device position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusPowerSaveRecoveryTime.**xfsCHKStatusManagedServiceName (19)
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. This value is zero if either the power saving mode has not been activated or no power save control is supported.

xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusAntiFraudModule.**xfsCHKStatusManagedServiceName (20)
It contains the state of the anti-fraud module. It is a numeric type field.

3.3.2 CHK Reset Device Complete: an example

As an example, the following variable binding list represents a Reset Device Complete trap (6, 305) generated as the result of a request to reset the device from the remote management station. The device in question has a managed service name “CheckReader1”.

xfsMIBRoot.3.1.3.13	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult)
	0 (resetExecuted)
xfsMIBRoot.3.1.3.2	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	“CheckReader1”
xfsMIBRoot.3.1.3.3	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	5 (WFS_SERVICE_CLASS_CHK)
xfsMIBRoot.3.1.3.4	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	“CHK”
xfsMIBRoot.3.1.3.5	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServ

	iceType)
	1 (WFS_CHK_TYPECHK)
xfsMIBRoot.3.1.3.6	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	“.1.3.6.1.4.1.16213.2.5”
xfsMIBRoot.3.1.3.7	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	“ABC Corp Check Reader”
xfsMIBRoot.3.1.3.8	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	“Best Devices Incorporated”
xfsMIBRoot.3.1.3.9	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	“1.10”
xfsMIBRoot.3.1.3.11	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	“20/03/2003 15:40:53 -300”
xfsMIBRoot.3.1.3.12	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	“1.23”
xfsMIBRoot.2.5.1.2.1.**3**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevice**.xfsCHKStatusManagedServiceName)
	1 (WFS_STAT_DEVONLINE)
xfsMIBRoot.2.5.1.2.1.**2**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusNumberSubDevices**.xfsCHKStatusManagedServiceName)
	0 (No sub device)
xfsMIBRoot.2.5.1.2.1.**4**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusMedia**.xfsCHKStatusManagedServiceName)
	7 (xfsCHKMediaRetained)
xfsMIBRoot.2.5.1.2.1.**5**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusInk**.xfsCHKStatusManagedServiceName)
	3 (xfsCHKInkLow)
xfsMIBRoot.2.5.1.2.1.**100**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusExtraStatus**.xfsCHKStatusManagedServiceName)
	'\0'\0' (No extra data)
xfsMIBRoot.2.5.1.2.1.**6**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusGuidanceCheckUnit**.xfsCHKStatusManagedServiceName)
	1 (value corresponding to WFS_CHK_GUIDANCE_OFF)
xfsMIBRoot.2.5.1.2.1.**7**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusDevicePosition**.xfsCHKStatusManagedServiceName)
	1 (xfsCHKDeviceInPosition)
xfsMIBRoot.2.5.1.2.1.**8**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusPowerSaveRecoveryTime**.xfsCHKStatusManagedServiceName)
	3 (3 seconds to recover from power saving mode)
xfsMIBRoot.2.5.1.2.1.**9**.Index	(xfsMIBRoot.xfsManagedService.xfsCHK.xfsCHKV1.xfsCHKStatusTable.xfsCHKStatusEntry.**xfsCHKStatusAntiFraudModule**.xfsCHKStatusManagedServiceName)
	2 (xfsCHKAFMOK)

4. Appendix A - CHK MIB sub-tree

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

4.1 CHK MIB in SMIv2 and SMIv1 ASN-1 format

SMIv2_xfsCHK.mib SMIv1_xfsCHK.mib

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

```
-- ******************************************************************
-- XFS 3.30 MIB for CHK
-- Management Information Base for XFS CHK Device
--
-- The CHK Number is 5
-- The ASN.1 prefix to, and including the CHK is: 1.3.6.1.4.1.16213.2.5
--
-- ******************************************************************

  XFS-CHK-MIB DEFINITIONS ::= BEGIN

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


--
-- Type definitions
--

-- ******************************************************************
-- CHK #defines
-- ******************************************************************
    IxfsCHKMediaStatus ::= INTEGER
      {
      xfsCHKMediaNotSupp(1),
      xfsCHKMediaNotPresent(2),
      xfsCHKMediaRequired(3),
      xfsCHKMediaPresent(4),
      xfsCHKMediaJammed(5),
      xfsCHKMediaEjected(6),
      xfsCHKMediaRetained(7)
      }

    IxfsCHKInkStatus ::= INTEGER
      {
      xfsCHKInkNotSupp(1),
      xfsCHKInkFull(2),
      xfsCHKInkLow(3),
      xfsCHKInkOut(4)
      }

    IxfsCHKDevicePositionStatus ::= INTEGER
      {
      xfsCIMDeviceInPosition(1),
      xfsCIMDeviceNotInPosition(2),
      xfsCIMDevicePosUnknown(3),
      xfsCIMDevicePosNotSupported(4)
      }

    IxfsCHKAntiFraudModuleStatus ::= INTEGER
      {
      xfsCHKAFMNotSupported(1),
      xfsCHKAFMOK(2),
```

```
      xfsCHKAFMInop(3),
      xfsCHKAFMDeviceDetected(4),
      xfsCHKAFMUnknown(5)
      }

    IxfsCHKCapabilitiesStamp ::= INTEGER
      {
      xfsCHKStampNone(2),
      xfsCHKStampFront(3),
      xfsCHKStampBack(4),
      xfsCHKStampBoth(5)
      }

    IxfsCHKCapabilitiesImageCapture ::= INTEGER
      {
      xfsCHKICapNone(2),
      xfsCHKICapFront(3),
      xfsCHKICapBack(4),
      xfsCHKICapBoth(5)
      }

    IxfsCHKCapabilitiesDeviceType ::= INTEGER { xfsCHKTypeChk(2) }



--
-- Node definitions
--

-- ******************************************************************
-- Version 1 of CHK MIB
--
-- The ASN.1 prefix to, and including the Version 1 of CHK is:
1.3.6.1.4.1.16213.2.5.1
--
-- ******************************************************************
    -- 1.3.6.1.4.1.16213.2.5.1
    xfsCHKV1 OBJECT IDENTIFIER ::= { xfsCHK 1 }


    -- 1.3.6.1.4.1.16213.2.5.1.1
    xfsCHKInstances OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Number that represents the number of CHK managed services."
      ::= { xfsCHKV1 1 }


-- ********************************************************************
-- CHK Device Status Table
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.5.1.2
    xfsCHKStatusTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsCHKStatusEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the CHK status table."
      ::= { xfsCHKV1 2 }


    -- 1.3.6.1.4.1.16213.2.5.1.2.1
    xfsCHKStatusEntry OBJECT-TYPE
      SYNTAX XfsCHKStatusEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "CHK Device Status Table Entry."
      INDEX { xfsCHKStatusManagedServiceName }
      ::= { xfsCHKStatusTable 1 }


    XfsCHKStatusEntry ::=
```

```
  SEQUENCE {
    xfsCHKStatusManagedServiceName
      DisplayString,
    xfsCHKStatusNumberSubDevices
      Integer32,
    xfsCHKStatusDevice
      IxfsMIBDeviceStatus,
    xfsCHKStatusMedia
      IxfsCHKMediaStatus,
    xfsCHKStatusInk
      IxfsCHKInkStatus,
    xfsCHKStatusGuidanceCheckUnit
      Integer32,
    xfsCHKStatusDevicePosition
      IxfsCHKDevicePositionStatus,
    xfsCHKStatusPowerSaveRecoveryTime
      Integer32,
    xfsCHKStatusAntiFraudModule
      IxfsCHKAntiFraudModuleStatus,
    xfsCHKStatusExtraStatus
      OCTET STRING
   }

-- 1.3.6.1.4.1.16213.2.5.1.2.1.1
xfsCHKStatusManagedServiceName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Instance identifier of the managed service."
  ::= { xfsCHKStatusEntry 1 }


-- 1.3.6.1.4.1.16213.2.5.1.2.1.2
xfsCHKStatusNumberSubDevices OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Number of sub devices supported by the CHK device."
  ::= { xfsCHKStatusEntry 2 }


-- 1.3.6.1.4.1.16213.2.5.1.2.1.3
xfsCHKStatusDevice OBJECT-TYPE
  SYNTAX IxfsMIBDeviceStatus
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Device status."
  ::= { xfsCHKStatusEntry 3 }


-- 1.3.6.1.4.1.16213.2.5.1.2.1.4
xfsCHKStatusMedia OBJECT-TYPE
  SYNTAX IxfsCHKMediaStatus
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Media Status.
    xfsCHKMediaNotSupp (1),
    xfsCHKMediaNotPresent (2),
    xfsCHKMediaRequired (3),
    xfsCHKMediaPresent (4),
    xfsCHKMediaJammed (5),
    xfsCHKMediaEjected (6),
    xfsCHKMediaRetained (7)"
  ::= { xfsCHKStatusEntry 4 }


-- 1.3.6.1.4.1.16213.2.5.1.2.1.5
xfsCHKStatusInk OBJECT-TYPE
  SYNTAX IxfsCHKInkStatus
  MAX-ACCESS read-only
  STATUS current
```

```
      DESCRIPTION
        "Contains the state of the ink.
        xfsCHKInkNotSupp(1),
        xfsCHKInkFull (2),
        xfsCHKInkLow (3),
        xfsCHKInkOut (4)"
      ::= { xfsCHKStatusEntry 5 }


    -- 1.3.6.1.4.1.16213.2.5.1.2.1.6
    xfsCHKStatusGuidanceCheckUnit OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "State of the guidance light indicator."
      ::= { xfsCHKStatusEntry 6 }


    -- 1.3.6.1.4.1.16213.2.5.1.2.1.7
    xfsCHKStatusDevicePosition OBJECT-TYPE
      SYNTAX IxfsCHKDevicePositionStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the device position.
        xfsCHKDeviceInPosition(1),
        xfsCHKDeviceNotInPosition(2),
        xfsCHKDevicePosUnknown(3),
        xfsCHKDevicePosNotSupported(4)."
      ::= { xfsCHKStatusEntry 7 }


    -- 1.3.6.1.4.1.16213.2.5.1.2.1.8
    xfsCHKStatusPowerSaveRecoveryTime OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the actual number of seconds required by the device to resume
its normal operational state from the current power saving mode. This value is zero
if either the power saving mode has not been activated or no power save control is
supported."
      ::= { xfsCHKStatusEntry 8 }


    -- 1.3.6.1.4.1.16213.2.5.1.2.1.9
    xfsCHKStatusAntiFraudModule OBJECT-TYPE
      SYNTAX IxfsCHKAntiFraudModuleStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the state of the anti-fraud module.
        xfsCHKAFMNotSupported(1),
        xfsCHKAFMOK(2),
        xfsCHKAFMInop(3),
        xfsCHKAFMDeviceDetected(4),
        xfsCHKAFMUnknown(5)."
      ::= { xfsCHKStatusEntry 9 }


    -- 1.3.6.1.4.1.16213.2.5.1.2.1.100
    xfsCHKStatusExtraStatus OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Vendor dependent additional device status information."
      ::= { xfsCHKStatusEntry 100 }

-- ********************************************************************
-- CHK Sub Device Status Table
--
-- Note that the CHK device does not currently have sub-devices. The
```

```
-- sub-device table is not required for this device and is shown as an
-- example for those devices that do support sub-devices.
--
-- Note, to ensure consistency across all MIB extensions OID 16213.2.5.1.3
-- must be reserved for the sub-device table.
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.5.1.3
    xfsCHKSubDeviceTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsCHKSubDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the CHK Sub-Device Status Table."
      ::= { xfsCHKV1 3 }


    -- 1.3.6.1.4.1.16213.2.5.1.3.1
    xfsCHKSubDeviceEntry OBJECT-TYPE
      SYNTAX XfsCHKSubDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "CHK Sub-Device Status Table Entry."
      INDEX { xfsCHKSubDeviceManagedServiceName, xfsCHKSubDeviceIndex }
      ::= { xfsCHKSubDeviceTable 1 }


    XfsCHKSubDeviceEntry ::=
      SEQUENCE {
        xfsCHKSubDeviceManagedServiceName
          DisplayString,
        xfsCHKSubDeviceIndex
          INTEGER
       }

--    As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
--   xfsCHKSubDeviceValue               INTEGER }
    -- 1.3.6.1.4.1.16213.2.5.1.3.1.1
    xfsCHKSubDeviceManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsCHKSubDeviceEntry 1 }


    -- 1.3.6.1.4.1.16213.2.5.1.3.1.2
    xfsCHKSubDeviceIndex OBJECT-TYPE
      SYNTAX INTEGER (1..65535)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Index into the array of sub devices supported."
      ::= { xfsCHKSubDeviceEntry 2 }


--    As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
-- xfsCHKSubDeviceValue  OBJECT-TYPE
--   SYNTAX        INTEGER
--   ACCESS        read-only
--   STATUS        mandatory
--   DESCRIPTION    "Returns the value of the sub device referenced by the index."
--   ::= {xfsCHKSubDeviceEntry 3}
-- ********************************************************************
-- CHK Error Table
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.5.1.4
    xfsCHKErrorTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsCHKErrorEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
```

```
        "Define the set of MIB Variables for the CHK Error Table."
      ::= { xfsCHKV1 4 }


    -- 1.3.6.1.4.1.16213.2.5.1.4.1
    xfsCHKErrorEntry OBJECT-TYPE
      SYNTAX XfsCHKErrorEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "CHK Error Table Entry."
      INDEX { xfsCHKErrorManagedServiceName, xfsCHKErrorCommandCode,
xfsCHKErrorResponseCode }
      ::= { xfsCHKErrorTable 1 }


    XfsCHKErrorEntry ::=
      SEQUENCE {
        xfsCHKErrorManagedServiceName
          DisplayString,
        xfsCHKErrorCommandCode
          INTEGER,
        xfsCHKErrorResponseCode
          INTEGER,
        xfsCHKErrorCount
          Integer32
       }

    -- 1.3.6.1.4.1.16213.2.5.1.4.1.1
    xfsCHKErrorManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsCHKErrorEntry 1 }


    -- 1.3.6.1.4.1.16213.2.5.1.4.1.2
    xfsCHKErrorCommandCode OBJECT-TYPE
      SYNTAX INTEGER (501..600)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The executable command code supported by the Service
        Provider associated with the error count of interest."
      ::= { xfsCHKErrorEntry 2 }


    -- 1.3.6.1.4.1.16213.2.5.1.4.1.3
    xfsCHKErrorResponseCode OBJECT-TYPE
      SYNTAX INTEGER (0..99 | 500..599)
      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."
      ::= { xfsCHKErrorEntry 3 }


    -- 1.3.6.1.4.1.16213.2.5.1.4.1.4
    xfsCHKErrorCount OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "The counter value corresponding to the managed service,
        command code and response code."
      ::= { xfsCHKErrorEntry 4 }


-- ***********************************************************************
-- CHK Reset Table
-- ***********************************************************************
```

```
    -- 1.3.6.1.4.1.16213.2.5.1.5
    xfsCHKResetTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsCHKResetEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Defines the set of MIB Variables for the CHK Reset Table."
      ::= { xfsCHKV1 5 }


    -- 1.3.6.1.4.1.16213.2.5.1.5.1
    xfsCHKResetEntry OBJECT-TYPE
      SYNTAX XfsCHKResetEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "CHK Reset Table Entry."
      INDEX { xfsCHKResetManagedServiceName }
      ::= { xfsCHKResetTable 1 }


    XfsCHKResetEntry ::=
      SEQUENCE {
        xfsCHKResetManagedServiceName
          DisplayString,
        xfsCHKResetAll
          Integer32,
        xfsCHKResetTimestamp
          DisplayString
       }

    -- 1.3.6.1.4.1.16213.2.5.1.5.1.1
    xfsCHKResetManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsCHKResetEntry 1 }


    -- 1.3.6.1.4.1.16213.2.5.1.5.1.2
    xfsCHKResetAll 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."
      ::= { xfsCHKResetEntry 2 }


    -- 1.3.6.1.4.1.16213.2.5.1.5.1.3
    xfsCHKResetTimestamp OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Date and time the last reset of the counters was
        performed."
      ::= { xfsCHKResetEntry 3 }


--  ********************************************************************
-- CHK Reset Device Table
-- ******************************************************************
    -- 1.3.6.1.4.1.16213.2.5.1.6
    xfsCHKResetDeviceTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsCHKResetDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the CHK Reset Device Table."
      ::= { xfsCHKV1 6 }
```

```
-- 1.3.6.1.4.1.16213.2.5.1.6.1
xfsCHKResetDeviceEntry OBJECT-TYPE
  SYNTAX XfsCHKResetDeviceEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "CHK Reset Device Table Entry."
  INDEX { xfsCHKResetDeviceManagedServiceName }
  ::= { xfsCHKResetDeviceTable 1 }


XfsCHKResetDeviceEntry ::=
  SEQUENCE {
    xfsCHKResetDeviceManagedServiceName
      DisplayString,
    xfsCHKResetDeviceAction
      INTEGER,
    xfsCHKResetDeviceMediaControl
      INTEGER,
    xfsCHKResetDeviceStatus
      INTEGER
   }

-- 1.3.6.1.4.1.16213.2.5.1.6.1.1
xfsCHKResetDeviceManagedServiceName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Instance identifier of the managed service."
  ::= { xfsCHKResetDeviceEntry 1 }


-- 1.3.6.1.4.1.16213.2.5.1.6.1.2
xfsCHKResetDeviceAction OBJECT-TYPE
  SYNTAX INTEGER { executeReset(1) }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Variable that initiates the device reset."
  ::= { xfsCHKResetDeviceEntry 2 }


-- 1.3.6.1.4.1.16213.2.5.1.6.1.3
xfsCHKResetDeviceMediaControl OBJECT-TYPE
  SYNTAX INTEGER
    {
    mediaDefault(1),
    mediaIn(2),
    mediaOut(3),
    mediaNoAction(4)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Variable that reports the media handling during the device reset."
  ::= { xfsCHKResetDeviceEntry 3 }


-- 1.3.6.1.4.1.16213.2.5.1.6.1.4
xfsCHKResetDeviceStatus OBJECT-TYPE
  SYNTAX INTEGER
    {
    resetIdle(1),
    resetInProgress(2)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Variable that reports the progress of the device reset."
  ::= { xfsCHKResetDeviceEntry 4 }

-- ***********************************************************************
```

```
-- CHK Device Capabilities Table
-- ***********************************************************************
    -- 1.3.6.1.4.1.16213.2.5.1.7
    xfsCHKCapabilitiesTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsCHKCapabilitiesEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the CHK capabilities table."
      ::= { xfsCHKV1 7 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1
    xfsCHKCapabilitiesEntry OBJECT-TYPE
      SYNTAX XfsCHKCapabilitiesEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "CHK Device Capabilities Table Entry."
      INDEX { xfsCHKCapabilitiesManagedServiceName }
      ::= { xfsCHKCapabilitiesTable 1 }


    XfsCHKCapabilitiesEntry ::=
      SEQUENCE {
        xfsCHKCapabilitiesManagedServiceName
          DisplayString,
        xfsCHKCapabilitiesDeviceType
          IxfsCHKCapabilitiesDeviceType,
        xfsCHKCapabilitiesCompoundDevice
          TruthValue,
        xfsCHKCapabilitiesMICR
          TruthValue,
        xfsCHKCapabilitiesOCR
          TruthValue,
        xfsCHKCapabilitiesAutofeed
          TruthValue,
        xfsCHKCapabilitiesEndorser
          TruthValue,
        xfsCHKCapabilitiesEncoder
          TruthValue,
        xfsCHKCapabilitiesStamp
          IxfsCHKCapabilitiesStamp,
        xfsCHKCapabilitiesImageCapture
          IxfsCHKCapabilitiesImageCapture,
        xfsCHKCapabilitiesFontNames
          OCTET STRING,
        xfsCHKCapabilitiesEncodeNames
          OCTET STRING,
        xfsCHKCapabilitiesCharSupport
          Integer32,
        xfsCHKCapabilitiesGuidanceCheckUnit
          Integer32,
        xfsCHKCapabilitiesPowerSaveControl
          TruthValue,
        xfsCHKCapabilitiesAntiFraudModule
          TruthValue,
        xfsCHKCapabilitiesSynchronizableCommands
          OCTET STRING,
        xfsCHKCapabilitiesExtraCapability
          OCTET STRING
       }

    -- 1.3.6.1.4.1.16213.2.5.1.7.1.1
    xfsCHKCapabilitiesManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsCHKCapabilitiesEntry 1 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.2
    xfsCHKCapabilitiesDeviceType OBJECT-TYPE
```

```
    SYNTAX IxfsCHKCapabilitiesDeviceType
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Type of CHK device."
    ::= { xfsCHKCapabilitiesEntry 2 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.3
  xfsCHKCapabilitiesCompoundDevice OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies if the logical device is part of a compound device."
    ::= { xfsCHKCapabilitiesEntry 3 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.4
  xfsCHKCapabilitiesMICR OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies if the logical device can read MICR on check."
    ::= { xfsCHKCapabilitiesEntry 4 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.5
  xfsCHKCapabilitiesOCR OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies if the logical device can read OCR on check."
    ::= { xfsCHKCapabilitiesEntry 5 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.6
  xfsCHKCapabilitiesAutofeed OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether device has autofeed or manual feed capability."
    ::= { xfsCHKCapabilitiesEntry 6 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.7
  xfsCHKCapabilitiesEndorser OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether programmer endorser present or absent."
    ::= { xfsCHKCapabilitiesEntry 7 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.8
  xfsCHKCapabilitiesEncoder OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether encoder present or absent."
    ::= { xfsCHKCapabilitiesEntry 8 }


  -- 1.3.6.1.4.1.16213.2.5.1.7.1.9
  xfsCHKCapabilitiesStamp OBJECT-TYPE
    SYNTAX IxfsCHKCapabilitiesStamp
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
```

```
        "Specifies the physical dimensions of the check where the endorser stamp
can be used."
      ::= { xfsCHKCapabilitiesEntry 9 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.10
    xfsCHKCapabilitiesImageCapture OBJECT-TYPE
      SYNTAX IxfsCHKCapabilitiesImageCapture
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the physical dimensions that can be image captured."
      ::= { xfsCHKCapabilitiesEntry 10 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.11
    xfsCHKCapabilitiesFontNames OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The names of the fonts supported for reading; each is terminated with a
null and the string is terminated with two nulls."
      ::= { xfsCHKCapabilitiesEntry 11 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.12
    xfsCHKCapabilitiesEncodeNames OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The names of the fonts supported for encoding; each is terminated with a
null and the string is terminated with two nulls."
      ::= { xfsCHKCapabilitiesEntry 12 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.13
    xfsCHKCapabilitiesCharSupport OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the physical dimensions that can be image captured."
      ::= { xfsCHKCapabilitiesEntry 13 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.14
    xfsCHKCapabilitiesGuidanceCheckUnit OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "It contains the capability of the guidelight. Allowed values are as
follows as a combination flag."
      ::= { xfsCHKCapabilitiesEntry 14 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.15
    xfsCHKCapabilitiesPowerSaveControl OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies whether power saving control is available."
      ::= { xfsCHKCapabilitiesEntry 15 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.16
    xfsCHKCapabilitiesAntiFraudModule OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies whether the anti-fraud module is available."
```

```
      ::= { xfsCHKCapabilitiesEntry 16 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.17
    xfsCHKCapabilitiesSynchronizableCommands OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the list of execute command IDs that can be synchronized."
      ::= { xfsCHKCapabilitiesEntry 17 }


    -- 1.3.6.1.4.1.16213.2.5.1.7.1.100
    xfsCHKCapabilitiesExtraCapability OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Pointer to a list of vendor-specific, or any other extended, information.
The information is returned as a series of 'key=value' strings so that it is easily
extensible by Service Providers.Each string is null-terminated, with the final
string terminating with two null characters."
      ::= { xfsCHKCapabilitiesEntry 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
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.3.0.105
    xfsCHKDetailedDSCTrap NOTIFICATION-TYPE
      OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapEvent,
        xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsCHKStatusDevice,
xfsCHKStatusNumberSubDevices, xfsCHKStatusMedia, xfsCHKStatusInk,
        xfsCHKStatusExtraStatus, xfsCHKStatusGuidanceCheckUnit,
xfsCHKStatusDevicePosition, xfsCHKStatusPowerSaveRecoveryTime,
xfsCHKStatusAntiFraudModule
         }
      STATUS current
      DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
      ::= { xfsTrapV2 105 }


    -- 1.3.6.1.4.1.16213.3.0.305
    xfsCHKResetDeviceCompleteTrap NOTIFICATION-TYPE
      OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsCHKStatusDevice, xfsCHKStatusNumberSubDevices,
xfsCHKStatusMedia, xfsCHKStatusInk, xfsCHKStatusExtraStatus,
        xfsCHKStatusGuidanceCheckUnit, xfsCHKStatusDevicePosition,
xfsCHKStatusPowerSaveRecoveryTime, xfsCHKStatusAntiFraudModule }
      STATUS current
      DESCRIPTION
        "This trap indicates the Reset action has complete and reports the
        state of the device after the reset."
      ::= { xfsTrapV2 305 }
```

```
  END

--
-- SMIv2_xfsCHK.mib
--
```

5. Appendix B - C-Header files

5.1 XFSMIBCHK.H

```
/**********************************************************************
*                                                                     *
* xfsmibchk.h      CEN/XFS - MIB CHK                                  *
*                                                                     *
*               Version 3.30  --  Feb 6, 2017                         *
*                                                                     *
**********************************************************************/

#ifndef __inc_xfsmibchk__h
#define __inc_xfsmibchk__h

#ifdef __cplusplus
extern "C" {
#endif

enum IxfsCHKMediaStatus
{
        xfsCHKMediaNotSupp          = 1,
        xfsCHKMediaNotPresent,
        xfsCHKMediaRequired,
        xfsCHKMediaPresent,
        xfsCHKMediaJammed,
        xfsCHKMediaEjected,
        xfsCHKMediaRetained,

} xfsCHKMediaStatus;


enum IxfsCHKInkStatus
{
        xfsCHKInkNotSupp            = 1,
        xfsCHKInkFull,
        xfsCHKInkLow,
        xfsCHKInkOut,

} xfsCHKInkStatus;

enum IxfsCHKDevicePositionStatus
{
        xfsCHKDeviceInPosition      = 1,
        xfsCHKDeviceNotInPosition,
        xfsCHKDevicePosUnknown,
        xfsCHKDevicePosNotSupported

} xfsCHKDevicePositionStatus;

enum IxfsCHKAntiFraudModuleStatus
{
    xfsCHKAFMNotSupported              = 1,
    xfsCHKAFMOK,
    xfsCHKAFMInop,
    xfsCHKAFMDeviceDetected,
    xfsCHKAFMUnknown

} xfsCHKAntiFraudModuleStatus;

enum IxfsCHKCapabilitiesStamp
{
        xfsCHKStampNone             = 2,
        xfsCHKStampFront,
        xfsCHKStampBack,
        xfsCHKStampBoth,

} xfsCHKCapabilitiesStamp;

enum IxfsCHKCapabilitiesImageCapture
{
```

```
        xfsCHKICapNone              = 2,
        xfsCHKICapFront,
        xfsCHKICapBack,
        xfsCHKICapBoth,

} xfsCHKCapabilitiesImageCapture;


/**********************************************************************
*                                                                     *
*       MIB Variables for the Status Table                            *
*                                                                     *
**********************************************************************/
#define        xfsCHKStatusManagedServiceName      (1)
#define        xfsCHKStatusNumberSubDevices        (2)
#define        xfsCHKStatusDevice                  (3)
#define        xfsCHKStatusMedia                   (4)
#define        xfsCHKStatusInk                     (5)
#define        xfsCHKStatusGuidanceCheckUnit       (6)
#define        xfsCHKStatusDevicePosition          (7)
#define        xfsCHKStatusPowerSaveRecoveryTime   (8)
#define        xfsCHKStatusAntiFraudModule         (9)

#define        xfsCHKStatusExtraStatus             (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        xfsCHKCapabilitiesManagedServiceName      (1)
#define        xfsCHKCapabilitiesDeviceType              (2)
#define        xfsCHKCapabilitiesCompoundDevice          (3)
#define        xfsCHKCapabilitiesMICR                    (4)
#define        xfsCHKCapabilitiesOCR                     (5)
#define        xfsCHKCapabilitiesAutofeed                (6)
#define        xfsCHKCapabilitiesEndorser                (7)
#define        xfsCHKCapabilitiesEncoder                 (8)
#define        xfsCHKCapabilitiesStamp                   (9)
#define        xfsCHKCapabilitiesImageCapture            (10)
#define        xfsCHKCapabilitiesFontNames               (11)
#define        xfsCHKCapabilitiesEncodeNames             (12)
#define        xfsCHKCapabilitiesCharSupport             (13)
#define        xfsCHKCapabilitiesGuidanceCheckUnit       (14)
#define        xfsCHKCapabilitiesPowerSaveControl        (15)
#define        xfsCHKCapabilitiesAntiFraudModule         (16)
#define        xfsCHKCapabilitiesSynchronizableCommands  (17)
#define        xfsCHKCapabilitiesExtraCapability        (100)

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

#endif  /* __inc_xfsmibchk__h */
```