

CEN

CWA 14050-34

WORKSHOP

July 2007

AGREEMENT

ICS 35.220; 35.240.15; 35.240.40

English version

**Extensions for Financial Services (XFS) interface specification -
Release 3.03 - Part 34: XFS MIB Device Specific Definitions -
Check Reader/Scanner Device Class MIB 1.1**

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

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

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

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

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



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

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

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

Ref. No.:CWA 14050-34:2007 D/E/F

Table of Contents

FOREWORD	3
1. INTRODUCTION	5
2. XFS CHK MIB VARIABLES	7
2.1 XFS CHK STATUS TABLE	7
2.1.1 <i>xfCHKStatusTable: States</i>	7
2.2 XFS CHK SUB DEVICE TABLE	8
2.3 XFS CHK ERROR TABLE	8
2.4 XFS CHK RESET TABLE	9
2.5 XFS CHK RESET DEVICE TABLE	9
3. CHK TRAPS	11
3.1 CHK DETAILED DEVICE STATUS CHANGE TRAP	11
3.1.1 <i>CHK Detailed Device Status Change Trap Format</i>	11
3.1.2 <i>CHK Detailed Device Status Change Trap: an example</i>	13
3.2 CHK SUB-DEVICE STATUS CHANGE TRAP	14
3.3 CHK RESET DEVICE COMPLETE TRAP	14
3.3.1 <i>CHK Reset Device Complete Trap Format</i>	14
3.3.2 <i>CHK Reset Device Complete: an example</i>	16
4. APPENDIX A - CHK MIB SUB-TREE	18
4.1 CHK MIB IN SMIV2 AND SMIV1 ASN-1 FORMAT	18
5. APPENDIX B - C-HEADER FILES	25
5.1 XFSMIBCHK.H	25

Foreword

This CWA is revision 3.03 of the XFS interface specification.

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

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

This document supersedes CWA 14050-34:2004.

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

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

Part 2: Service Classes Definition; Programmer's Reference

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Part 44: XFS MIB Application Management

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

Revision History:

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

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

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

1. Introduction

This document provides the device specific MIB definition (Management Information Base) variables for the 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:

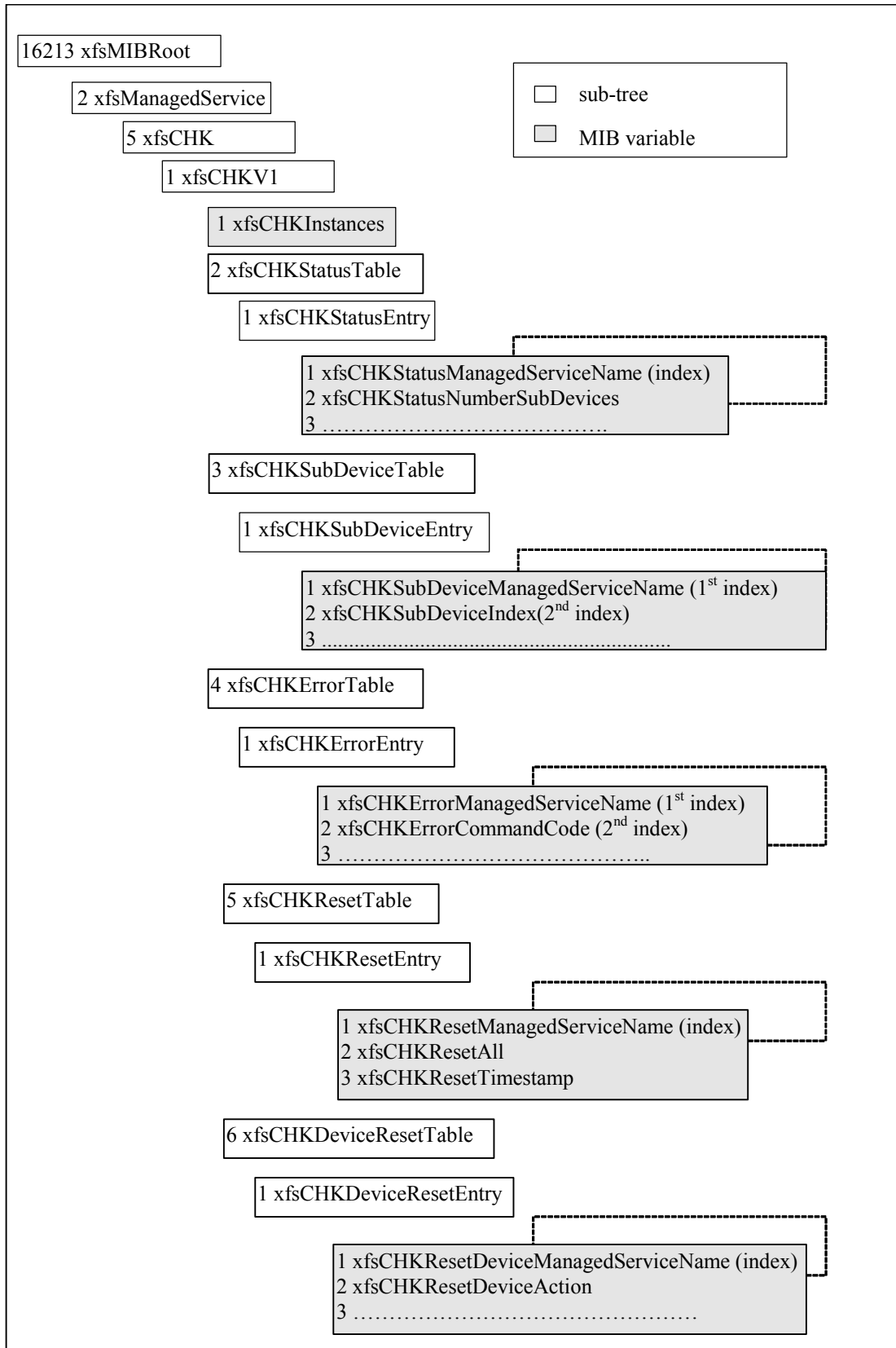
xfsMIBRoot

- 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.

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.



Section 2 describes how the Status, Sub-Device, Error and Reset 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 `xfCHKStatusTable` groups the variables identifying device status information, statistics and auxiliary variables. It is indexed through a single parameter, `xfCHKStatusManagedServiceName`. 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.

`xfCHKStatusManagedServiceName` 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 `xfCHKStatusMedia(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:

`xfMIBRoot.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.

`xfCHKStatusManagedServiceName (1)`

Uniquely identifies the managed service.

`xfCHKStatusNumberSubDevices (2)`

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

`xfCHKStatusDevice (3)`

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

Value	Meaning
<code>xfCHKDevOnline (1)</code>	The device is on-line (i.e., powered on and operable).

xfsCHKDevOffline (2)	The device is off-line (e.g., the operator has taken the device offline by turning a switch or pulling out the device).
xfsCHKDevPowerOff (3)	The device is powered off or physically not connected.
xfsCHKDevNoDevice (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.
xfsCHKDevHWError (5)	The device is inoperable due to a hardware error.
xfsCHKDevUserError (6)	The device is inoperable because a person is preventing proper device operation.
xfsCHKDevBusy (7)	The device is busy and unable to process an execute command at this time.

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.

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.

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* 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:

- *xfCHKErrorManagedServiceName(1)* which provides the primary index to the service in question. It is Display String field. The *xfCHKErrorManagedServiceName* parameter corresponds to the value of *xfMIBRoot.xfGeneral.xfMIBV1.xfManagedServiceTable.xfManagedServiceEntry.xfManagedServiceName* in the general table. E.g. “CheckReader1”.
- *xfCHKErrorCommandCode(2)* is an index which identifies the command code that that response code related to, e.g. WFS_CMD_CHK_PROCESS_FORM (501). It is a 32 bit numerical field.
- *xfCHKErrorResponseCode(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
- *xfCHKErrorCount(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:

```
xfMIBRoot.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 *xfCHKResetTable* contains the *xfCHKResetAll* and *xfCHKResetTimestamp* variables and is indexed by the single variable, *xfCHKResetManagedServiceName*. When the *xfCHKResetAll* 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 *xfCHKResetTable* is defined as:

- *xfCHKResetManagedServiceName(1)* which provides the index to the service in question. It is Display String field. The *xfCHKResetManagedServiceName* parameter corresponds to the value of *xfMIBRoot.xfGeneral.xfMIBV1.xfManagedServiceTable.xfManagedServiceEntry.xfManagedServiceName* in the general table. E.g. “CheckReader1”.
- *xfCHKResetAll(2)* is a read-write variable. Issue of a Set command on the *xfCHKResetAll* 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 *xfCHKResetAll* variable will return 0 (zero).
- *xfCHKResetTimestamp(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 *xfCHKResetAll* variable represented by:

```
xfMIBRoot.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 *xfCHKResetDeviceTable(6)* is indexed by the single variable, *xfCHKResetDeviceManagedServiceName*. This table contains variables which monitor and control the execution of the reset request.

The *xfCHKResetDeviceAction* 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 *xfCHKResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request

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

The *xfCHKResetDeviceTable*(6) is defined as:

- *xfCHKResetDeviceManagedServiceName*(1) which provides the index to the service in question. It is a Display String field. The *xfCHKResetDeviceManagedServiceName* parameter corresponds to the value of *xfMIBRoot.xfGeneral.xfMIBV1.xfManagedServiceTable.xfManagedServiceEntry.xfManagedServiceName* in the general table. E.g. "CheckReader1".
- *xfCHKResetDeviceAction*(2) is a read-write variable. Issue of a Set command on the *xfCHKResetDeviceAction* variable with value *executeReset*(1) will result in the device being reset as described above.
- *xfCHKResetDeviceMediaControl*(3) is a read-only variable. This variable reports how any media found within the device is handled. The value of the *xfCHKResetDeviceMediaControl* 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.
- *xfCHKResetDeviceStatus*(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 *xfCHKResetDeviceAction* variable represented by:

`xfMIBRoot.2.5.1.6.1.2.12.67.104.101.99.107.82.101.97.100.101.114.49`

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 an CHK Detailed Device Status Change trap.

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

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

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

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

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

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

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" (No extra data)

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

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

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

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

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

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

xfsxfsCHKStatusDevice.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.

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 a 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.xfsCommonTrapManagedServiceClasses)
	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.3.1.3.11	(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)

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

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



SMIV2\xfsCHK.mib

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



SMIV1\xfsCHK.mib

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

```
--*****
-- XFS 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
            FROM SNMPv2-TC
        xfsCHK, xfsTrap, IxfsMIBDeviceStatus
            FROM XFSMIB;

-- *****
-- 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)
    }

-- *****
-- 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
-- *****
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 }

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,
        xfsCHKStatusExtraStatus
            OCTET STRING
    }

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

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

xfsCHKStatusDevice OBJECT-TYPE
    SYNTAX IxfsMIBDeviceStatus
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Device status."

```

```

 ::= { xfsCHKStatusEntry 3 }

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 }

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 }

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.
-- *****
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 }

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

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
-- *****
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 }

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
    }

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

```

```

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 }

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 }

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
-- *****
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 }

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
    }

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

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 }

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
-- *****
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 }

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
}

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

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

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 }

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 }

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

-- *****
-- Trap definitions
-- *****
xfsCHKDetailedDSCTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
              xfsCommonTrapManagedServiceClass,
              xfsCommonTrapManagedServiceClassName,
              xfsCommonTrapManagedServiceType,
              xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
              xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion,
              xfsCommonTrapEvent,
              xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsCHKStatusDevice,
              xfsCHKStatusNumberSubDevices, xfsCHKStatusMedia,
              xfsCHKStatusInk, xfsCHKStatusExtraStatus }
    STATUS current
    DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
    ::= { xfsTrapV2 105 }

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

```

END

5. Appendix B - C-Header files

5.1 XFSMIBCHK.H



xfsmibchk.h

```

/*****
*
* xfsmibchk.h      WOSA/XFS - MIB CHK counters
*
*      Version 1.00  --  Jan 20, 2004
*
*****/

#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;

/*****
*
*      MIB Variables for the Status Table
*
*****/
#define      xfsCHKStatusManagedServiceName      (1)
#define      xfsCHKStatusNumberSubDevices        (2)
#define      xfsCHKStatusDevice                  (3)
#define      xfsCHKStatusMedia                   (4)
#define      xfsCHKStatusInk                     (5)
#define      xfsCHKStatusExtraStatus             (100)

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

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

#endif /* __inc_xfsmibchk_h */

```