# CEN

# WORKSHOP

# AGREEMENT

# CWA 16374-39

September 2014

**ICS** 35.200; 35.240.15; 35.240.40

English version

# Extensions for Financial Services (XFS) interface specification - Release 3.20 - Part 39: XFS MIB Device Specific Definitions - Alarm Device Class MIB 3.20



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

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



Ref. No.:CWA 16374-39:2014 E

# Table of Contents

# Foreword

This CWA is revision 3.20 of the XFS interface specification.

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties on 2011-06-29, the constitution of which was supported by CEN following the public call for participation made on 1998-06-24. The specification is continuously reviewed and commented in the CEN/ISSS Workshop on XFS. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this revision 3.20.

A list of the individuals and organizations which supported the technical consensus represented by the CEN Workshop Agreement is available to purchasers from the CEN-CENELEC Management Centre. These organizations were drawn from the banking sector. The CEN/ISSS XFS Workshop gathered suppliers as well as banks and other financial service companies.

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

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

Part 2: Service Class Definition - Programmer's Reference

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Parts 19 - 28: Reserved for future use.

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

Part 29: XFS MIB Architecture and SNMP Extensions MIB 3.20

Part 30: XFS MIB Device Specific Definitions - Printer Device Class MIB 3.20

Part 31: XFS MIB Device Specific Definitions - Identification Card Device Class MIB 3.20

Part 32: XFS MIB Device Specific Definitions - Cash Dispenser Device Class MIB 3.20

Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class MIB 3.20

Part 34: XFS MIB Device Specific Definitions - Check Reader/Scanner Device Class MIB 3.20

Part 35: XFS MIB Device Specific Definitions - Depository Device Class MIB 3.20

Part 36: XFS MIB Device Specific Definitions - Text Terminal Unit Device Class MIB 3.20

Part 37: XFS MIB Device Specific Definitions - Sensors and Indicators Unit Device Class MIB 3.20

Part 38: XFS MIB Device Specific Definitions - Camera Device Class MIB 3.20

Part 39: XFS MIB Device Specific Definitions - Alarm Device Class MIB 3.20

Part 40: XFS MIB Device Specific Definitions - Card Embossing Unit Class MIB 3.20

Part 41: XFS MIB Device Specific Definitions - Cash-In Module Device Class MIB 3.20

Part 42: Reserved for future use.

Part 43: XFS MIB Device Specific Definitions - Vendor Dependent Mode Class MIB 3.20

Part 44: XFS MIB Application Management MIB 3.20

Part 45: XFS MIB Device Specific Definitions - Card Dispenser Device Class MIB 3.20

Part 46: XFS MIB Device Specific Definitions - Barcode Reader Device Class MIB 3.20

Part 47: XFS MIB Device Specific Definitions - Item Processing Module Device Class MIB 3.20

Parts 48 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 62: Printer and Scanning Device Class Interface Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.10 ( CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 75: Card Dispenser Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 76: Barcode Reader Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

Part 77: Item Processing Module Device Class Interface - Migration from Version 3.10 (CWA 15748) to Version 3.20 (this CWA) - Programmer's Reference

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is available online from http://www.cen.eu/cen/Sectors/Sectors/ISSS/Activity/Pages/WSXFS.aspx.

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

The final review/endorsement round for parts 29-47 of this CWA was started on 2014-06-23 and was successfully closed on 2014-07-23. The final text for parts 29-47 of this CWA was submitted to CEN for publication on 2014-08-22.

This CEN Workshop Agreement is publicly available as a reference document from the National Members of The following countries: 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. |

# 1. Introduction

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

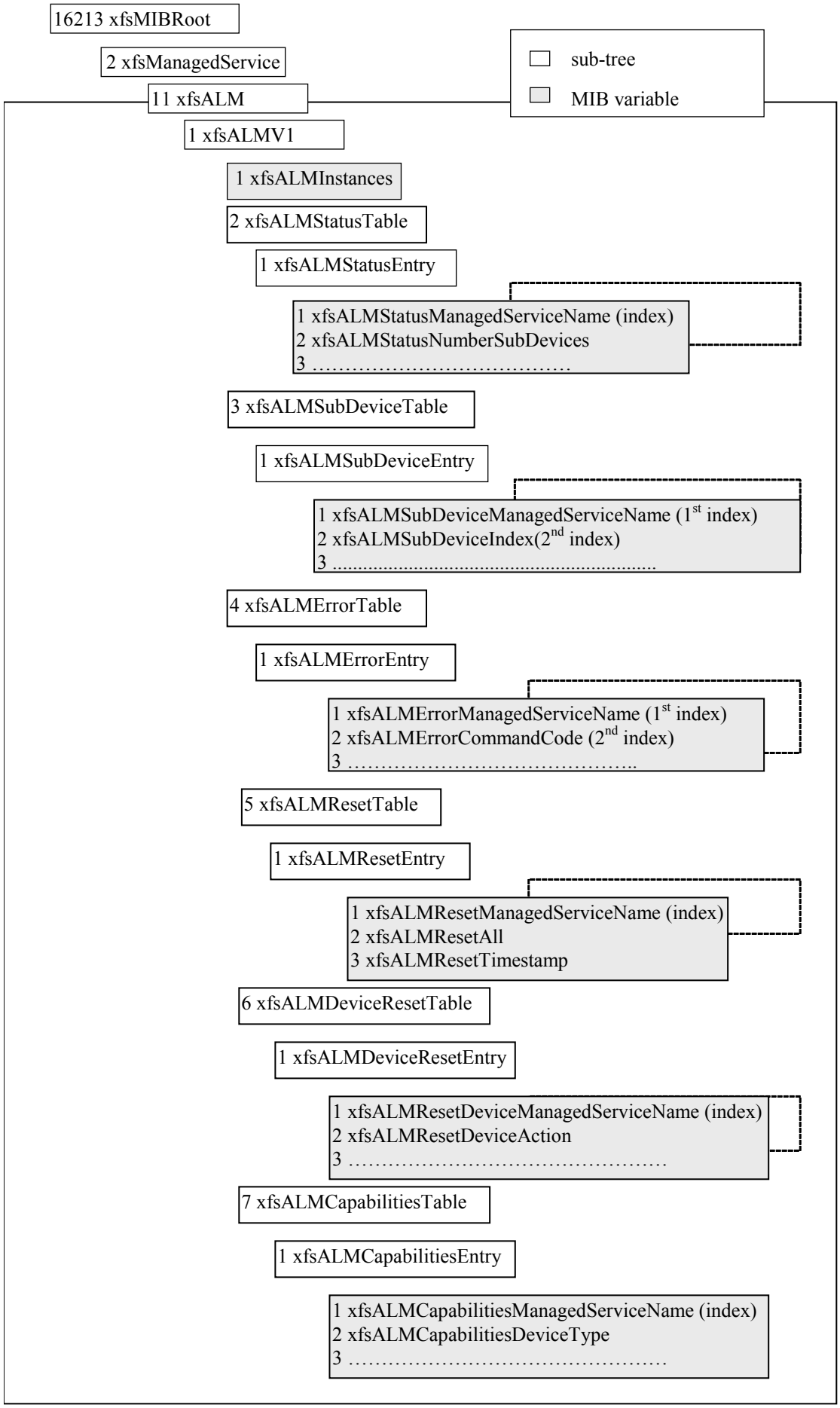
xfsMIBRoot

- xfsManagedService (2)
  - xfsALM (11)
    - xfsALMV1 (1)

The xfsALMV1 sub-tree contains the following variables:

- *xfsALMInstances(1)* is the number of managed services for the ALM class installed on the XFS subsystem. It is a 32 bit numerical field.
- *xfsALMStatusTable(2)* identifies the table for the ALM variables.
- *xfsALMSubDevicesTable(3)* not applicable to the ALM device.
- *xfsALMErrorTable(4)* identifies the table for ALM error counters.
- *xfsALMResetTable(5)* identifies the table for the ALM reset variable.
- *xfsALMResetDeviceTable(6)* identifies the table for the ALM reset device variables.
- *xfsALMCapabilitiesTable(7)* identifies the table for the ALM 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 *xfsALMV1* sub-tree.


16213 xfsMIBRoot
2 xfsManagedService
11 xfsALM
1 xfsALMV1
sub-tree
MIB variable
1 xfsALMInstances
2 xfsALMStatusTable
1 xfsALMStatusEntry
1 xfsALMStatusManagedServiceName (index)
2 xfsALMStatusNumberSubDevices
3 ……
3 xfsALMSubDeviceTable
1 xfsALMSubDeviceEntry
1 xfsALMSubDeviceManagedServiceName (1st index)
2 xfsALMSubDeviceIndex(2nd index)
3 .....
4 xfsALMErrorTable
1 xfsALMErrorEntry
1 xfsALMErrorManagedServiceName (1st index)
2 xfsALMErrorCommandCode (2nd index)
3 ……..
5 xfsALMResetTable
1 xfsALMResetEntry
1 xfsALMResetManagedServiceName (index)
2 xfsALMResetAll
3 xfsALMResetTimestamp
6 xfsALMDeviceResetTable
1 xfsALMDeviceResetEntry
1 xfsALMResetDeviceManagedServiceName (index)
2 xfsALMResetDeviceAction
3 ……
7 xfsALMCapabilitiesTable
1 xfsALMCapabilitiesEntry
1 xfsALMCapabilitiesManagedServiceName (index)
2 xfsALMCapabilitiesDeviceType
3 ……

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

# 2. XFS ALM MIB variables

This section describes the MIB variables for the tables of the ALM Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Alarm Device Class Interface Programmer's Reference.*
The following are some general notes pertaining to the MIB variables:

- All command response counters maintained by the Service Provider are persistent across re-boots.
- One application command may trigger only one command-related counter to be updated.
- One application command may trigger one or multiple status variables to be updated.
- All command response counters are read-writable unless otherwise specified.
- Each managed service has a Reset table that allows all the response counters to be reset.
- Each managed service has a Reset Device table that allows the WFS_CMD_ALM_RESET command to be executed from the management station.

## 2.1 XFS ALM Status Table

The *xfsALMStatusTable(2)* groups the variables identifying device status information, statistics and additional variables. It is indexed through a single parameter, *xfsALMStatusManagedServiceName*. 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.

*xfsALMStatusManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the ALM class.

As an example, the identifier for the device status value of *xfsALMStatusAlarmSet(4)* for a device with managed service name equal to "Alarm1" is as follows:

| Character | A | l | a | r | m | 1 |
|---|---|---|---|---|---|---|
| ASCII Hex | 41 | 6C | 61 | 72 | 6D | 31 |
| ASCII Dec | 65 | 108 | 97 | 114 | 109 | 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.2.11.1.2.1.4.6.65.108.97.114.109.49*

### 2.1.1 xfsALMStatusTable: States

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

xfsALMStatusManagedServiceName (1)
Uniquely identifies the managed service.

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

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

| Value | Meaning |
|---|---|
| xfsDevOnline(1) | The device is present, powered on and online (i.e., operational, not busy processing a request and not in an error state). |

| | |
|---|---|
| xfsDevOffline(2) | The device is offline (e.g., the operator has taken the device offline by turning a switch or pulling out the device). |
| xfsDevPowerOff(3) | The device is powered off or physically not connected. |
| xfsDevNoDevice(4) | There is no device intended to be there; e.g. this type of self service machine does not contain such a device or it is internally not configured . |
| xfsDevHWError(5) | The device is present but inoperable due to a hardware fault that prevents it from being used. |
| xfsDevUserError(6) | The device is present but a person is preventing proper device operation. The application should suspend the device operation or remove the device from service until the Service Provider generates a device state change event indicating the condition of the device has changed e.g. the error is removed (WFS_ALM_DEVONLINE) or a permanent error condition has occurred (WFS_ALM_DEVHWERROR). |
| xfsDevBusy(7) | The device is busy and unable to process an execute command at this time. |
| xfsDevFraudAttempt (8) | The device is present but is inoperable because it has detected a fraud attempt. |
| xfsDevPotentialFraud (9) | The device has detected a potential fraud attempt and is capable of remaining in service. |

xfsALMStatusAlarmSet (4)

Specifies the state of the alarm as either Reset (FALSE) or Set (TRUE). It is a TruthValue (RFC1253-MIB) where 1 = TRUE and 2 = FALSE.

xfsALMStatusAntiFraudModule (5)

Specifies the state of the anti-fraud module as one of the following values:

| Value | Meaning |
|---|---|
| xfsALMAFMNotSupported (1) | No anti-fraud module is available. |
| xfsALMAFMOK (2) | Anti-fraud module is in a good state and no foreign device is detected. |
| xfsALMAFMInop (3) | Anti-fraud module is inoperable. |
| xfsALMAFMDeviceDetected (4) | Anti-fraud module detected the presence of a foreign device. |
| xfsALMAFMUnknown (5) | The state of the anti-fraud module cannot be determined. |

xfsALMStatusExtraStatus (100)

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

## 2.2 XFS ALM Sub Device Table

The ALM service class does not support any sub-devices, therefore the *xfsALMStatusNumberSubDevices* 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 ALM Error Table

The *xfsALMErrorTable(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:

*xfsALMErrorManagedServiceName*
*xfsALMErrorCommandCode*
*xfsALMErrorResponseCode*

The *xfsALMErrorTable* is defined as:

- *xfsALMErrorManagedServiceName(1*) which provides the primary index to the service in question. It is Display String field. The *xfsALMErrorManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. “Alarm1”.
- *xfsALMErrorCommandCode(2)* is an index which identifies the command code that that response code is related to. It is a 32 bit numerical field.
- *xfsALMErrorResponseCode(3*) is an index which identifies the response code that the count is required for. It is the absolute value of the error code. It is a 32 bit numerical field.
- *xfsALMErrorCount(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_INTERNAL_ERROR (-15) error returned from the WFS_CMD_ALM_RESET_ALARM (1102) command for a device with managed service name equal to “Alarm1” is as follows:

*xfsMIBRoot.2.11.1.4.1.4.6.65.108.97.114.109.49.1102.15*

## 2.4 XFS ALM Reset Table

The *xfsALMResetTable(5)* contains the *xfsALMResetAll* and *xfsALMResetTimestamp* variables and is indexed by the single variable, *xfsALMResetManagedServiceName*. When the *xfsALMResetAll* 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 *xfsALMResetTable* is defined as:

- *xfsALMResetManagedServiceName(1)* which provides the index to the service in question. It is Display String field. The *xfsALMResetManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneralxfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. “Alarm1”.
- *xfsALMResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsALMResetAll* 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 *xfsALMResetAll* variable will return 0 (zero).
- *xfsALMResetTimestamp(3)* is a read-only variable which represents the UTC date and time when the counters in the error table was reset, it is a Display String field. The data is formatted in the following way: "DD/MM/YYYY HH:MM:SS +ZZZ" where DD/MM/YYYY HH:MM:SS is the local date and time. ZZZ is the bias, which is the difference, in minutes, between Co-ordinated Universal Time (UTC) and local time.

As an example, all the error counts can be reset for a device with managed service name equal to “Alarm1” by setting the value zero in the *xfsALMResetAll* variable represented by:

*xfsMIBRoot.2.11.1.5.1.2.6.65.108.97.114.109.49*

## 2.5 XFS ALM Reset Device Table

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

The *xfsALMResetDeviceAction* 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_ALM_RESET command will be issued.
4. Exclusive access to the device will be relinquished when the WFS_CMD_ALM_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 *xfsALMResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request.

The *xfsALMResetDeviceTable* is defined as:

- *xfsALMResetDeviceManagedServiceName(1)* which provides the index to the service in question. It is a Display String field. The *xfsALMResetDeviceManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "Alarm1".
- *xfsALMResetDeviceAction(2)* is a read-write variable. Issue of a Set command on the *xfsALMResetDeviceAction* variable with value *executeReset(1)* will result in the device being reset as described above.
- *xfsALMResetDeviceMediaControl(3)* is a read-only variable. As there is no media in the ALM device class this variable can only report the *mediaDefault* value.
- *xfsALMResetDeviceStatus(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 "Alarm1" is reset by setting the *xfsALMResetDeviceAction* variable represented by:

*xfsMIBRoot.2.11.1.6.1.2.6.65.108.97.114.109.49*

## 2.6 XFS ALM Capabilities Table

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

*xfsALMCapabilitiesManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the ALM class.
As an example, the identifier for the device capabilities value of *xfsALMCapabilitiesProgrammaticallyDeactivate(2)* for a device with managed service name equal to "Alarm1" is as follows:

| Character | A | l | a | r | m | 1 |
|---|---|---|---|---|---|---|
| ASCII Hex | 41 | 6C | 61 | 72 | 6D | 31 |
| ASCII Dec | 65 | 108 | 97 | 114 | 109 | 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.2.11.1.7.1.2.6.65.108.97.114.109.49*

### 2.6.1 xfsALMCapabilitiesTable: Capabilities

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

xfsALMCapabilitiesManagedServiceName (1)
Uniquely identifies the managed service.

xfsALMCapabilitiesProgrammaticallyDeactivate (2)
Specifies whether the Alarm can be programmatically deactivated not in a TruthValue variable as follows:

| Value | Meaning |
|---|---|
| True(1) | The device can be programmatically deactivated. |
| False(2) | The device cannot be programmatically deactivated. |

xfsALMCapabilitiesAntiFraudModule (3)
Specifies whether the anti-fraud module is available or not 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. |

xfsALMCapabilitiesExtraCapability (100)
Contains vendor dependent additional device capability information as an OCTET STRING. The information is returned as a series of "*key=value*" strings. Each string is null-terminated, with the final string terminating with two null characters.

# 3. ALM Traps

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

## 3.1 ALM 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 ALM 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 ALM reflect the ALM 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_ALM_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 111 defines the trap as an ALM 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 ALM Detailed Device Status Change Trap Format

The following defines the variable bindings included in the ALM 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 is zero as this device class does not have a type.

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 ALM MIB class is represented by .1.3.6.1.4.1.16213.2.11

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.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusDevice**.xfsALMStatusManagedServiceName (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.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusNumberSubDevices.**xfsALMStatusManagedServiceName (14)
Defines how many sub-devices the service has. This is zero for this device class.

xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAlarmSet.**xfsALMStatusManagedServiceName (15)
It specifies the state of the Alarm as either Reset (False) or Set (True). It is a Truth Value where 1 = TRUE and 2 = FALSE.

xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusExtraStatus.**xfsALMStatusManagedServiceName (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.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAntiFraudModule.**xfsALMStatusManagedServiceName (17)
Specifies the state of the anti-fraud module. It is a numeric type field.

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

As an example, the following variable binding list represents a detailed device status change trap (6, 111) that is generated for an ALM with a managed service name of "Alarm1". It reports that the device is in HARDWARE ERROR.

| | |
|---|---|
| xfsMIBRoot.3.1.3.1 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName) |
| | "SST System 1" |
| | |
| xfsMIBRoot.3.1.3.2 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName) |
| | "Alarm1" |
| | |
| xfsMIBRoot.3.1.3.3 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass) |
| | 11 (WFS_SERVICE_CLASS_ALM) |
| | |
| xfsMIBRoot.3.1.3.4 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName) |
| | "ALM" |
| | |
| xfsMIBRoot.3.1.3.5 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType) |
| | 0 |
| | |
| xfsMIBRoot.3.1.3.6 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid) |
| | ".1.3.6.1.4.1.16213.2.11" |
| | |
| xfsMIBRoot.3.1.3.7 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName) |
| | "JC Alarm Systems" |
| | |
| xfsMIBRoot.3.1.3.8 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor) |
| | "JC Alarms 1066" |
| | |
| 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.2.11.1.2.1.**3**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusDevice**.xfsALMStatusManagedServiceName) |
| | 5 (WFS_STAT_HWERROR) |
| | |
| xfsMIBRoot.2.11.1.2.1.**2**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusNumberSubDevices**.xfsALMStatusManagedServiceName) |
| | 0 (No sub device) |
| | |
| xfsMIBRoot.2.11.1.2.1.**4**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAlarmSet.**xfsALMStatusManagedServiceName) |
| | 2 (FALSE) |
| | |
| xfsMIBRoot.2.11.1.2.1.**100**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusExtraStatus**.xfsALMStatusManagedServiceName) |
| | '\0''\0' ( No extra data ) |
| | |

| | |
|---|---|
| xfsMIBRoot.2.11.1.2.1.**5**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAntiFraudModule.**xfsALMStatusManagedServiceName) |
| | 2 (xfsALMAFMOK) |

## 3.2 ALM Sub-Device Status Change Trap

The ALM does not currently support any sub-devices so the ALM Sub-Device Status Change Trap is not currently defined. The SNMP Specific trap value 211 is reserved in case a sub-device is ever added to the ALM device class.

## 3.3 ALM Reset Device Complete Trap

On the ALM 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 311 defines the trap as an ALM Reset Device Complete trap.

### 3.3.1 ALM Reset Device Complete Trap Format

The following defines the variable bindings included in the ALM 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 **xfsALMStatusDevice** 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 is zero as this device class does not have a type.

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 ALM MIB class is represented by .1.3.6.1.4.1.16213.2.11

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.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusDevice**.xfsALMStatusManagedServiceName (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.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusNumberSubDevices.**xfsALMStatusManagedServiceName (13)
Defines how many sub-devices the service has. This is zero for this device class.

xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAlarmSet.**xfsALMStatusManagedServiceName (14)
It specifies the state of the Alarm as either Reset (False) or Set (True). It is a Truth Value where 1 = TRUE and 2 = FALSE.

xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusExtraStatus.**xfsALMStatusManagedServiceName (15)
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.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAntiFraudModule.**xfsALMStatusManagedServiceName (16)
Specifies the state of the anti-fraud module. It is a numeric type field.

### 3.3.2 ALM Reset Device Complete: an example

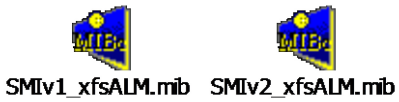
As an example, the following variable binding list represents a Reset Device Complete trap (6, 311) generated as a result of a request to reset the device from the remote management station. The device in question has a managed service name “Alarm1”.

| | |
|---|---|
| xfsMIBRoot.3.1.3.13 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult) |
| | 0 (resetExecuted) |
| | |
| xfsMIBRoot.3.1.3.2 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName) |
| | “Alarm1” |
| | |
| xfsMIBRoot.3.1.3.3 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass) |
| | 11 (WFS_SERVICE_CLASS_ALM) |
| | |
| xfsMIBRoot.3.1.3.4 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName) |
| | “ALM” |
| | |
| xfsMIBRoot.3.1.3.5 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType) |
| | 0 |
| | |
| xfsMIBRoot.3.1.3.6 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid) |
| | “.1.3.6.1.4.1.16213.2.11” |
| | |
| xfsMIBRoot.3.1.3.7 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName) |
| | “JC Alarm Systems” |
| | |
| xfsMIBRoot.3.1.3.8 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor) |
| | “JC Alarms 1066” |
| | |
| 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.2.11.1.2.1.**3**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusDevice**.xfsALMStatusManagedServiceName) |
| | 1 (WFS_STAT_ONLINE) |
| | |
| xfsMIBRoot.2.11.1.2.1.**2**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusNumberSubDevices**.xfsALMStatusManagedServiceName) |
| | 0 (No sub device) |
| | |
| xfsMIBRoot.2.11.1.2.1.**4**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAlarmSet.**xfsALMStatusManagedServiceName) |
| | 2 (FALSE) |
| | |
| xfsMIBRoot.2.11.1.2.1.**100**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusExtraStatus**.xfsALMStatusManagedServiceName) |
| | '\0"\0' ( No extra data ) |
| | |
| xfsMIBRoot.2.11.1.2.1.**5**.Index | (xfsMIBRoot.xfsManagedService.xfsALM.xfsALMV1.xfsALMStatusTable.xfsALMStatusEntry.**xfsALMStatusAntiFraudModule.**xfsALMStatusManagedServiceName) |
| | 2 (xfsALMAFMOK) |

# 4. Appendix A - ALM MIB sub-tree

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

## 4.1 ALM MIB in SMIv2 and SMIv1 format




***The following text is the content of xfsALM.mib in SMIv2 format.***

```
-- ****************************************************************
-- XFS 3.20 MIB for ALM
-- Management Information Base for XFS ALM Device
--
-- The ALM Number is 11
-- The ASN.1 prefix to, and including the ALM is: 1.3.6.1.4.1.16213.2.11
--
-- ****************************************************************

  XFS-ALM-MIB DEFINITIONS ::= BEGIN

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




--
-- Node definitions
--

-- ****************************************************************
-- ALM Status #defines
-- ****************************************************************
        IxfsALMAntiFraudModuleStatus ::= INTEGER
    {
      xfsALMAFMNotSupported(1),
      xfsALMAFMOK(2),
      xfsALMAFMInop(3),
      xfsALMAFMDeviceDetected(4),
      xfsALMAFMUnknown(5)
    }


-- ****************************************************************
-- Version 1 of ALM MIB
--
-- The ASN.1 prefix to, and including the Version 1 of ALM is:
1.3.6.1.4.1.16213.2.11.1
--
-- ****************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1
    xfsALMV1 OBJECT IDENTIFIER ::= { xfsALM 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.1
    xfsALMInstances OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
```

```
        "Number that represents the number of ALM managed services."
      ::= { xfsALMV1 1 }


-- ************************************************************************
-- ALM Device Status Table
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1.2
    xfsALMStatusTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsALMStatusEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the ALM status table."
      ::= { xfsALMV1 2 }


    -- 1.3.6.1.4.1.16213.2.11.1.2.1
    xfsALMStatusEntry OBJECT-TYPE
      SYNTAX XfsALMStatusEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "ALM Device Status Table Entry."
      INDEX { xfsALMStatusManagedServiceName }
      ::= { xfsALMStatusTable 1 }


    XfsALMStatusEntry ::=
      SEQUENCE {
        xfsALMStatusManagedServiceName
          DisplayString,
        xfsALMStatusNumberSubDevices
          Integer32,
        xfsALMStatusDevice
          IxfsMIBDeviceStatus,
        xfsALMStatusAlarmSet
          TruthValue,
        xfsALMStatusAntiFraudModule
          IxfsALMAntiFraudModuleStatus,
        xfsALMStatusExtraStatus
          OCTET STRING
       }

    -- 1.3.6.1.4.1.16213.2.11.1.2.1.1
    xfsALMStatusManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsALMStatusEntry 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.2.1.2
    xfsALMStatusNumberSubDevices OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Number of sub devices supported by the ALM device."
      ::= { xfsALMStatusEntry 2 }


    -- 1.3.6.1.4.1.16213.2.11.1.2.1.3
    xfsALMStatusDevice OBJECT-TYPE
      SYNTAX IxfsMIBDeviceStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Device status."
```

```
      ::= { xfsALMStatusEntry 3 }


    -- 1.3.6.1.4.1.16213.2.11.1.2.1.4
    xfsALMStatusAlarmSet OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Alarm Set."
      ::= { xfsALMStatusEntry 4 }


    -- 1.3.6.1.4.1.16213.2.11.1.2.1.5
    xfsALMStatusAntiFraudModule OBJECT-TYPE
      SYNTAX IxfsALMAntiFraudModuleStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "It contains the state of the anti-fraud module. Allowed values are:
        xfsALMAFMNotSupported(1),
        xfsALMAFMOK(2),
        xfsALMAFMInop(3),
        xfsALMAFMDeviceDetected(4),
        xfsALMAFMUnknown(5)."
      ::= { xfsALMStatusEntry 5 }


    -- 1.3.6.1.4.1.16213.2.11.1.2.1.100
    xfsALMStatusExtraStatus OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Vendor dependent additional device status information."
      ::= { xfsALMStatusEntry 100 }


-- ************************************************************************
-- ALM Sub Device Status Table
--
-- Note that the ALM 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 16213.2.11.1.3
-- must be reserved for the sub-device table.
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1.3
    xfsALMSubDeviceTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsALMSubDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the ALM Sub-Device Status Table."
      ::= { xfsALMV1 3 }


    -- 1.3.6.1.4.1.16213.2.11.1.3.1
    xfsALMSubDeviceEntry OBJECT-TYPE
      SYNTAX XfsALMSubDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "ALM Sub-Device Status Table Entry."
      INDEX { xfsALMSubDeviceManagedServiceName, xfsALMSubDeviceIndex }
      ::= { xfsALMSubDeviceTable 1 }


    XfsALMSubDeviceEntry ::=
      SEQUENCE {
```

```
        xfsALMSubDeviceManagedServiceName
          DisplayString,
        xfsALMSubDeviceIndex
          INTEGER
       }

--   As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
--   xfsALMSubDeviceValue INTEGER }
    -- 1.3.6.1.4.1.16213.2.11.1.3.1.1
    xfsALMSubDeviceManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsALMSubDeviceEntry 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.3.1.2
    xfsALMSubDeviceIndex OBJECT-TYPE
      SYNTAX INTEGER (1..65535)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Index into the array of sub devices supported."
      ::= { xfsALMSubDeviceEntry 2 }


--   As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
-- xfsALMSubDeviceValue   OBJECT-TYPE
-- SYNTAX       INTEGER
-- MAX-ACCESS   read-only
-- STATUS       current
-- DESCRIPTION  "Returns the value of the sub device referenced by the index."
-- ::= {xfsALMSubDeviceEntry 3}
-- ************************************************************************
-- ALM Error Table
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1.4
    xfsALMErrorTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsALMErrorEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the ALM Error Table."
      ::= { xfsALMV1 4 }


    -- 1.3.6.1.4.1.16213.2.11.1.4.1
    xfsALMErrorEntry OBJECT-TYPE
      SYNTAX XfsALMErrorEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "ALM Error Table Entry."
      INDEX { xfsALMErrorManagedServiceName, xfsALMErrorCommandCode,
xfsALMErrorResponseCode }
      ::= { xfsALMErrorTable 1 }


    XfsALMErrorEntry ::=
      SEQUENCE {
        xfsALMErrorManagedServiceName
          DisplayString,
        xfsALMErrorCommandCode
          INTEGER,
        xfsALMErrorResponseCode
          INTEGER,
        xfsALMErrorCount
```

```
        Integer32
      }

    -- 1.3.6.1.4.1.16213.2.11.1.4.1.1
    xfsALMErrorManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsALMErrorEntry 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.4.1.2
    xfsALMErrorCommandCode OBJECT-TYPE
      SYNTAX INTEGER (1101..1200)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The executable command code supported by the Service
        Provider associated with the error count of interest."
      ::= { xfsALMErrorEntry 2 }


    -- 1.3.6.1.4.1.16213.2.11.1.4.1.3
    xfsALMErrorResponseCode OBJECT-TYPE
      SYNTAX INTEGER (0..99 | 1100..1199)
      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."
      ::= { xfsALMErrorEntry 3 }


    -- 1.3.6.1.4.1.16213.2.11.1.4.1.4
    xfsALMErrorCount OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "The counter value corresponding to the managed service,
        command code and response code."
      ::= { xfsALMErrorEntry 4 }


-- ********************************************************************
-- ALM Reset Table
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1.5
    xfsALMResetTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsALMResetEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Defines the set of MIB Variables for the ALM Reset Table."
      ::= { xfsALMV1 5 }


    -- 1.3.6.1.4.1.16213.2.11.1.5.1
    xfsALMResetEntry OBJECT-TYPE
      SYNTAX XfsALMResetEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "ALM Reset Table Entry."
      INDEX { xfsALMResetManagedServiceName }
      ::= { xfsALMResetTable 1 }
```

```
    XfsALMResetEntry ::=
      SEQUENCE {
        xfsALMResetManagedServiceName
          DisplayString,
        xfsALMResetAll
          Integer32,
        xfsALMResetTimestamp
          DisplayString
       }

    -- 1.3.6.1.4.1.16213.2.11.1.5.1.1
    xfsALMResetManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsALMResetEntry 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.5.1.2
    xfsALMResetAll 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."
      ::= { xfsALMResetEntry 2 }


    -- 1.3.6.1.4.1.16213.2.11.1.5.1.3
    xfsALMResetTimestamp OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Date and time the last reset of the counters was performed."
      ::= { xfsALMResetEntry 3 }


--  ********************************************************************
-- ALM Reset Device Table
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1.6
    xfsALMResetDeviceTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsALMResetDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the ALM Reset Device Table."
      ::= { xfsALMV1 6 }


    -- 1.3.6.1.4.1.16213.2.11.1.6.1
    xfsALMResetDeviceEntry OBJECT-TYPE
      SYNTAX XfsALMResetDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "ALM Reset Device Table Entry."
      INDEX { xfsALMResetDeviceManagedServiceName }
      ::= { xfsALMResetDeviceTable 1 }


    XfsALMResetDeviceEntry ::=
      SEQUENCE {
        xfsALMResetDeviceManagedServiceName
          DisplayString,
        xfsALMResetDeviceAction
          INTEGER,
```

```
        xfsALMResetDeviceMediaControl
          INTEGER,
        xfsALMResetDeviceStatus
          INTEGER
       }

    -- 1.3.6.1.4.1.16213.2.11.1.6.1.1
    xfsALMResetDeviceManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsALMResetDeviceEntry 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.6.1.2
    xfsALMResetDeviceAction OBJECT-TYPE
      SYNTAX INTEGER { executeReset(1) }
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "Variable that initiates the device reset"
      ::= { xfsALMResetDeviceEntry 2 }


    -- 1.3.6.1.4.1.16213.2.11.1.6.1.3
    xfsALMResetDeviceMediaControl OBJECT-TYPE
      SYNTAX INTEGER { mediaDefault(1) }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Variable that reports the media handling during the device reset"
      ::= { xfsALMResetDeviceEntry 3 }


    -- 1.3.6.1.4.1.16213.2.11.1.6.1.4
    xfsALMResetDeviceStatus OBJECT-TYPE
      SYNTAX INTEGER
        {
        resetIdle(1),
        resetInProgress(2)
        }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Variable that reports the progress of the device reset"
      ::= { xfsALMResetDeviceEntry 4 }


--  ************************************************************************
-- ALM Device Capabilities Table
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.2.11.1.7
    xfsALMCapabilitiesTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsALMCapabilitiesEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the ALM capabilities table."
      ::= { xfsALMV1 7 }


    -- 1.3.6.1.4.1.16213.2.11.1.7.1
    xfsALMCapabilitiesEntry OBJECT-TYPE
      SYNTAX XfsALMCapabilitiesEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "ALM Device Capabilities Table Entry."
      INDEX { xfsALMCapabilitiesManagedServiceName }
```

```
      ::= { xfsALMCapabilitiesTable 1 }


    XfsALMCapabilitiesEntry ::=
      SEQUENCE {
        xfsALMCapabilitiesManagedServiceName
          DisplayString,
        xfsALMCapabilitiesProgrammaticallyDeactivate
          TruthValue,
        xfsALMCapabilitiesAntiFraudModule
          TruthValue,
        xfsALMCapabilitiesExtraCapability
          OCTET STRING
       }

    -- 1.3.6.1.4.1.16213.2.11.1.7.1.1
    xfsALMCapabilitiesManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsALMCapabilitiesEntry 1 }


    -- 1.3.6.1.4.1.16213.2.11.1.7.1.2
    xfsALMCapabilitiesProgrammaticallyDeactivate OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies if the Alarm can be programmatically deactivated."
      ::= { xfsALMCapabilitiesEntry 2 }


    -- 1.3.6.1.4.1.16213.2.11.1.7.1.3
    xfsALMCapabilitiesAntiFraudModule OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "This TruthValue variable specifies whether or not an anti-fraud module is
available."
      ::= { xfsALMCapabilitiesEntry 3 }


   -- 1.3.6.1.4.1.16213.2.11.1.7.1.100
    xfsALMCapabilitiesExtraCapability OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Vendor dependent additional device capablities information."
      ::= { xfsALMCapabilitiesEntry 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
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.3.0.111
    xfsALMDetailedDSCTrap NOTIFICATION-TYPE
```

```
      OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapEvent,
        xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsALMStatusDevice,
xfsALMStatusNumberSubDevices, xfsALMStatusAlarmSet,
        xfsALMStatusExtraStatus, xfsALMStatusAntiFraudModule }
      STATUS current
      DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
      ::= { xfsTrapV2 111 }


    -- 1.3.6.1.4.1.16213.3.0.311
    xfsALMResetDeviceCompleteTrap NOTIFICATION-TYPE
      OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsALMStatusDevice, xfsALMStatusNumberSubDevices,
xfsALMStatusAlarmSet, xfsALMStatusExtraStatus, xfsALMStatusAntiFraudModule
         }
      STATUS current
      DESCRIPTION
        "This trap indicates the Reset action has complete and reports the
        state of the device after the reset."
      ::= { xfsTrapV2 311 }



  END

--
-- SMIv2_xfsALM.MIB
--
```

# 5. Appendix B - C-Header files

## 5.1 XFSMIBALM.H

```
/******************************************************************************
*                                                                             *
* xfsmibalm.h      CEN/XFS - MIB ALM                                          *
*                                                                             *
*               Version 3.20  --  Mar 28, 2014                                *
*                                                                             *
******************************************************************************/

#ifndef __inc_xfsmibalm__h
#define __inc_xfsmibalm__h

#ifdef __cplusplus
extern "C" {
#endif


/*********************************************************************
*  ALM Status #defines
*********************************************************************/

enum IxfsALMAntiFraudModuleStatus
{
    xfsALMAFMNotSupported       = 1,
    xfsALMAFMOK,
    xfsALMAFMInop,
    xfsALMAFMDeviceDetected,
    xfsALMAFMUnknown

} xfsALMAntiFraudModuleStatus;


/******************************************************************************
*                                                                             *
*       MIB Variables for the Status Table                                    *
*                                                                             *
******************************************************************************/
#define  xfsALMStatusManagedServiceName      (1)
#define  xfsALMStatusNumberSubDevices        (2)
#define  xfsALMStatusDevice                  (3)
#define  xfsALMStatusAlarmSet                (4)
#define  xfsALMStatusAntiFraudModule         (5)
#define  xfsALMStatusExtraStatus             (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  xfsALMCapabilitiesManagedServiceName          (1)
#define  xfsALMCapabilitiesProgrammaticallyDeactivate  (2)
#define  xfsALMCapabilitiesAntiFraudModule             (3)
#define  xfsALMCapabilitiesExtraCapability             (100)


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

```
#endif

#endif  /* __inc_xfsmibalm__h */
```