

CEN

WORKSHOP

AGREEMENT

CWA 16926-43

February 2017

35.240.40

English version

**Extensions for Financial Services (XFS) interface specification –
Release 3.30 – Part 43: XFS MIB Device Specific Definitions –
Vendor Dependent Mode Class MIB 3.30**

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

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

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

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

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



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

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

© 2017 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.
Ref. No. CWA 16926-43:2017 D/E/F

Table of Contents

Foreword	3
1. Introduction	7
2. XFS VDM MIB variables	10
2.1 XFS VDM STATUS TABLE	10
2.1.1 <i>xfsVDMStatusTable: States</i>	10
2.2 XFS VDM SUB DEVICE TABLE	11
2.3 XFS VDM ERROR TABLE	11
2.4 XFS VDM RESET TABLE	11
2.5 XFS VDM RESET DEVICE TABLE	12
2.6 XFS VDM CAPABILITIES TABLE	12
2.6.1 <i>xfsVDMCapabilitiesTable: Capabilities</i>	12
3. VDM Traps	13
3.1 VDM DETAILED DEVICE STATUS CHANGE TRAP	13
3.1.1 <i>VDM Detailed Device Status Change Trap Format</i>	13
3.1.2 <i>VDM Detailed Device Status Change Trap: an example</i>	15
3.2 VDM SUB-DEVICE STATUS CHANGE TRAP	16
3.3 VDM RESET DEVICE COMPLETE TRAP	16
4. Appendix A - VDM MIB sub-tree	17
4.1 VDM MIB IN SMIV2 AND SMIV1 ASN-1 FORMAT	17
5. Appendix B - C-Header files	24
5.1 XFSMIBVDM.H	24

Foreword

This CWA is revision 3.30 of the XFS interface specification.

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

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

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

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

Part 2: Service Classes Definition - Programmer's Reference

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Parts 19 - 28: Reserved for future use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Part 42: Reserved for future use.

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

Part 44: XFS MIB Application Management

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

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

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

Parts 48 - 60 are reserved for future use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is available online from <http://www.cen.eu/work/areas/ict/ebusiness/pages/ws-xfs.aspx>.

The information in this document represents the Workshop's current views on the issues discussed as of the date

of publication. It is furnished for informational purposes only and is subject to change without notice. CEN makes no warranty, express or implied, with respect to this document.

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

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

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

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

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

Revision History:

1.10	April 15, 2007	Initial release of XFS Vendor Dependent Mode specification to coincide with XFS MIB version 1.1.
3.10	December 14, 2010	Update of the MIB to add support for a Capabilities table and to align the MIB with XFS 3.10.
3.20	March 28, 2014	Update release to align the MIB with XFS 3.20.
3.30	February 6, 2017	Update release to align the MIB with XFS 3.30.

1. Introduction

This document provides the device specific MIB definition (Management Information Base) variables for the xfsVDM sub-tree version 1.1, 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 xfsVDM version one sub-tree is identified by:

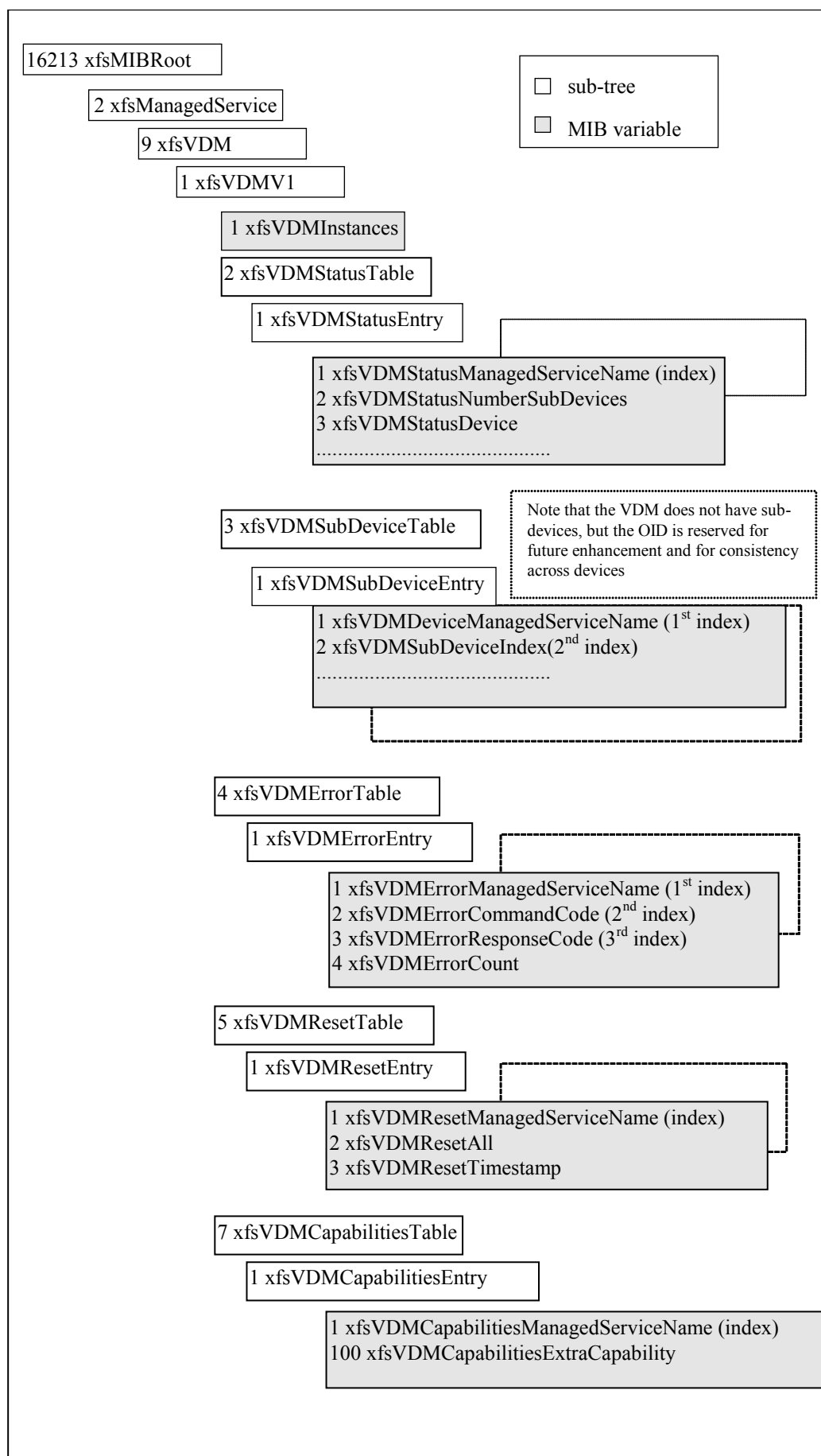
xfsMIBRoot

- xfsManagedService (2)
 - xfsVDM (9)
 - xfsVDMV1 (1)

The xfsVDMV1 sub-tree contains the following variables:

- * *xfsVDMInstances(1)* is the number of managed services for the VDM class installed on the XFS subsystem. It is a 32 bit numerical field.
- * *xfsVDMStatusTable(2)* identifies the table for the VDM variables.
- * *xfsVDMSubDeviceTable(3)* not applicable to the VDM device.
- * *xfsVDMErrorTable(4)* identifies the table for the VDM error counters.
- * *xfsVDMResetTable(5)* identifies the table for the VDM reset variable.
- * *xfsVDMCapabilitiesTable(7)* identifies the table for VDM 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 *xfsVDMV1* sub-tree.



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

2. XFS VDM MIB variables

This section describes the MIB variables for the tables of the VDM Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Vendor Dependent Mode 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.

2.1 XFS VDM Status Table

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

xfsVDMStatusManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the VDM class.

As an example, the identifier for the device status value of *xfsVDMStatusDevice(3)* for a device with managed service name equal to "VendorModel" is as follows:

Character	V	e	n	d	o	r	M	o	d	e	l
ASCII Hex	56	65	6E	64	6F	72	4D	6F	64	65	31
ASCII Dec	86	101	110	100	111	114	77	111	100	101	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.2.9.1.2.1.3.11.86.101.110.100.111.114.77.111.100.101.49

2.1.1 xfsVDMStatusTable: States

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

xfsVDMStatusManagedServiceName (1)
Uniquely identifies the managed service.

xfsVDMStatusNumberSubDevices (2)
Defines how many sub-devices the service has. This is always 0 (zero) in the VDM.

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

Value	Meaning
<i>xfsDevOnline</i> (1)	Vendor Dependent Mode service available.
<i>xfsDevOffline</i> (2)	Vendor Dependent Mode service unavailable.

xfsVDMStatusService (4)
It contains the state of the Vendor Dependent Mode service class. It is a numeric type field. Allowed values are:

Value	Meaning
<i>xfsVDMServiceEnterPending</i> (1)	The service is entering request pending.
<i>xfsVDMServiceActive</i> (2)	The service is active.
<i>xfsVDMServiceExitPending</i> (3)	The service is exiting request pending.
<i>xfsVDMServiceInactive</i> (4)	The service is inactive.

xfsvdmStatusExtraStatus (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 may be indicated by either a NULL pointer or a pointer to two consecutive null characters.

2.2 XFS VDM Sub Device Table

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

The *xfsvdmErrorTable*(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

xfsvdmErrorManagedServiceName
xfsvdmErrorCommandCode
xfsvdmErrorResponseCode

The *xfsvdmErrorTable* is defined as:

- *xfsvdmErrorManagedServiceName*(1) which provides the primary index to the service in question. It is Display String field. The *xfsvdmErrorManagedServiceName* parameter corresponds to the value of *xfsmibRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table. E.g. "VendorModel".
- *xfsvdmErrorCommandCode*(2) is an index which identifies the command code that that response code is related to, e.g. WFS_CMD_VDM_ENTER_MODE_REQ (901). It is a 32 bit numerical field.
- *xfsvdmErrorResponseCode*(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_INVALID_DATA (-52) is represented by 52. It is a 32 bit numerical field.
- *xfsvdmErrorCount*(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_INVALID_DATA (-52) error returned from the WFS_CMD_VDM_ENTER_MODE_REQ (901) command for a device with managed service name equal to "VendorModel" is as follows:

xfsmibRoot.2.9.1.4.1.4.11.86.101.110.100.111.114.77.111.100.101.49.901.52

2.4 XFS VDM Reset Table

The *xfsvdmResetTable*(5) contains the *xfsvdmResetAll* and *xfsvdmResetTimestamp* variables and is indexed by the single variable, *xfsvdmResetManagedServiceName*. When the *xfsvdmResetAll* 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 *xfsVDMResetTable(5)* is defined as:

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

xfsMIBRoot.2.9.1.5.1.2.11.86.101.110.100.111.114.77.111.100.101.49

2.5 XFS VDM Reset Device Table

The VDM service class does not support a device reset, therefore the VDM device reset table is not defined.

2.6 XFS VDM Capabilities Table

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

xfsVDMCapabilitiesManagedServiceName is the instance identifier of the managed service and uniquely identifies one instance of the VDM class.

As an example, the identifier for the device status value of *xfsVDMCapabilitiesExtraCapability(100)* for a device with managed service name equal to "VendorModel" is as follows:

Character	V	e	n	d	o	r	M	o	d	e	l
ASCII Hex	56	65	6E	64	6F	72	4D	6F	64	65	31
ASCII Dec	86	101	110	100	111	114	77	111	100	101	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.2.9.1.7.1.100.11.86.101.110.100.111.114.77.111.100.101.49

2.6.1 xfsVDMCapabilitiesTable: Capabilities

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

xfsVDMCapabilitiesManagedServiceName (1)

Uniquely identifies the managed service.

xfsVDMCapabilitiesExtraCapability (100)

It contains the vendor dependent additional device capabilities 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 may be indicated by either a NULL pointer or a pointer to two consecutive null characters.

3. VDM Traps

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

3.1 VDM 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 VDM 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 VDM reflect the VDM 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_VDM_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 109 defines the trap as a VDM Detailed Device Status Change trap.

3.1.1 VDM Detailed Device Status Change Trap Format

The following defines the variable bindings included in the VDM 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). This value 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 VDM MIB class is represented by .1.3.6.1.4.1.16213.2.9

`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.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusDevice.xfsVDMStatusManagedServiceName (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.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusNumberSubDevices.xfsVDMStatusManagedServiceName (14)

Defines how many sub-devices the service has. This value is always 0 (zero) for the VDM.

xfsMIBRoot.xfsManagedService.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusService.xfsVDMStatusManagedServiceName (15)

It contains the state of the Vendor Dependent Mode service. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusExtraStatus.xfsVDMStatusManagedServiceName (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.

3.1.2 VDM Detailed Device Status Change Trap: an example

As an example, the following variable binding list represents a detailed device status change trap (6, 109) that is generated for a VDM with a managed service name of “VendorModel”. It reports that the vendor mode is active.

xfMIBRoot.3.1.3.1	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName)
	“SST System 1”
xfMIBRoot.3.1.3.2	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName)
	“VendorModel”
xfMIBRoot.3.1.3.3	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass)
	9 (WFS_SERVICE_CLASS_VDM)
xfMIBRoot.3.1.3.4	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName)
	“VDM”
xfMIBRoot.3.1.3.5	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType)
	0 (zero)
xfMIBRoot.3.1.3.6	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid)
	“.1.3.6.1.4.1.16213.2.9”
xfMIBRoot.3.1.3.7	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName)
	“ABC Corp Vendor Mode”
xfMIBRoot.3.1.3.8	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor)
	“Best Devices Incorporated”
xfMIBRoot.3.1.3.9	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	“1.10”
xfMIBRoot.3.1.3.10	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent)
	4 (WFS_SYSE_DEVICE_STATUS)
xfMIBRoot.3.1.3.11	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	“20/03/2003 15:40:53 -300”
xfMIBRoot.3.1.3.12	(xfMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion)
	“1.23”
xfMIBRoot.2.9.1.2.1.3.Index	(xfMIBRoot.xfsManagedService.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusDevice.xfsVDMStatusManagedServiceName)
	1 (WFS_STAT_DEVONLINE)
xfMIBRoot.2.9.1.2.1.2.Index	(xfMIBRoot.xfsManagedService.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusNumberSubDevices.xfsVDMStatusManagedServiceName)
	0 (No sub device)
xfMIBRoot.2.9.1.2.1.4.Index	(xfMIBRoot.xfsManagedService.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusService.xfsVDMStatusManagedServiceName)
	2 (xfsVDMServiceActive)
xfMIBRoot.2.9.1.2.1.100.Index	(xfMIBRoot.xfsManagedService.xfsVDM.xfsVDMV1.xfsVDMStatusTable.xfsVDMStatusEntry.xfsVDMStatusExtraStatus.xfsVDMStatusManagedServiceName)
	“0” (No extra data)

3.2 VDM Sub-Device Status Change Trap

The VDM does not currently support any sub-devices so the VDM Sub-Device Status Change Trap is not currently defined. The SNMP Specific trap value 209 is reserved in case a sub-device is ever added to the VDM device class.

3.3 VDM Reset Device Complete Trap

The VDM does not currently support the reset device request so the VDM Reset Device Complete Trap is not currently defined. The SNMP Specific trap value 309 is reserved in case a reset device request is ever added to the VDM device class.

4. Appendix A - VDM MIB sub-tree

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

4.1 VDM MIB in SMIv2 and SMIv1 ASN-1 format



SMIv1_xfsVDM.mib



SMIv2_xfsVDM.mib

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

```
-- *****
-- XFS 3.30 MIB for VDM
-- Management Information Base for XFS VDM Device
--
-- The VDM Number is 9
-- The ASN.1 prefix to, and including the VDM is: 1.3.6.1.4.1.16213.2.9
--
-- *****

XFS-VDM-MIB DEFINITIONS ::= BEGIN

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

--
-- Type definitions
--

-- *****
-- VDM #defines
-- *****
IxfsVDMStatusService ::= INTEGER
{
    xfsVDMStatusServiceEnterPending(1),
    xfsVDMStatusServiceActive(2),
    xfsVDMStatusServiceExitPending(3),
    xfsVDMStatusServiceInactive(4)
}

--
-- Node definitions
--

-- *****
-- Version 1 of VDM MIB
--
-- The ASN.1 prefix to, and including the Version 1 of VDM is:
1.3.6.1.4.1.16213.2.9.1
--
-- *****
-- 1.3.6.1.4.1.16213.2.9.1
xfsVDMV1 OBJECT IDENTIFIER ::= { xfsVDM 1 }

-- 1.3.6.1.4.1.16213.2.9.1.1
```

```

xfsVDMInstances OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number that represents the number of VDM managed services."
    ::= { xfsVDMV1 1 }

-- *****
-- VDM Device Status Table
-- *****
-- 1.3.6.1.4.1.16213.2.9.1.2
xfsVDMStatusTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsVDMStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the VDM status table."
    ::= { xfsVDMV1 2 }

-- 1.3.6.1.4.1.16213.2.9.1.2.1
xfsVDMStatusEntry OBJECT-TYPE
    SYNTAX XfsVDMStatusEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "VDM Device Status Table Entry."
    INDEX { xfsVDMStatusManagedServiceName }
    ::= { xfsVDMStatusTable 1 }

XfsVDMStatusEntry ::=
    SEQUENCE {
        xfsVDMStatusManagedServiceName
            DisplayString,
        xfsVDMStatusNumberSubDevices
            Integer32,
        xfsVDMStatusDevice
            IxfsMIBDeviceStatus,
        xfsVDMStatusService
            IxfsVDMStatusService,
        xfsVDMStatusExtraStatus
            OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.9.1.2.1.1
xfsVDMStatusManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsVDMStatusEntry 1 }

-- 1.3.6.1.4.1.16213.2.9.1.2.1.2
xfsVDMStatusNumberSubDevices OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Number of sub devices supported by the VDM device."
    ::= { xfsVDMStatusEntry 2 }

-- 1.3.6.1.4.1.16213.2.9.1.2.1.3
xfsVDMStatusDevice OBJECT-TYPE
    SYNTAX IxfsMIBDeviceStatus
    MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION
    "Device status."
::= { xfsVDMStatusEntry 3 }

-- 1.3.6.1.4.1.16213.2.9.1.2.1.4
xfsVDMStatusService OBJECT-TYPE
    SYNTAX IxfsVDMStatusService
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Service Status.
        xfsVDMStatusServiceEnterPending(1),
        xfsVDMStatusServiceActive(2),
        xfsVDMStatusServiceExitPending(3),
        xfsVDMStatusServiceInactive(4)."
```

```

::= { xfsVDMStatusEntry 4 }

-- 1.3.6.1.4.1.16213.2.9.1.2.1.100
xfsVDMStatusExtraStatus OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Vendor dependent additional device status information."
::= { xfsVDMStatusEntry 100 }

-- *****
-- VDM Sub Device Status Table
--
-- Note that the VDM 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 16213.2.9.1.3
-- must be reserved for the sub-device table.
-- *****
-- 1.3.6.1.4.1.16213.2.9.1.3
xfsVDMSubDeviceTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsVDMSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the VDM status table."
::= { xfsVDMV1 3 }

-- 1.3.6.1.4.1.16213.2.9.1.3.1
xfsVDMSubDeviceEntry OBJECT-TYPE
    SYNTAX XfsVDMSubDeviceEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "VDM Sub-Device Status Table Entry."
    INDEX { xfsVDMSubDeviceManagedServiceName, xfsVDMSubDeviceIndex }
::= { xfsVDMSubDeviceTable 1 }

XfsVDMSubDeviceEntry ::=
    SEQUENCE {
        xfsVDMSubDeviceManagedServiceName
            DisplayString,
        xfsVDMSubDeviceIndex
            INTEGER
    }

-- As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
```

```

-- xfsVDMSubDeviceValue          INTEGER }
-- 1.3.6.1.4.1.16213.2.9.1.3.1.1
xfsVDMSubDeviceManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsVDMSubDeviceEntry 1 }

-- 1.3.6.1.4.1.16213.2.9.1.3.1.2
xfsVDMSubDeviceIndex OBJECT-TYPE
    SYNTAX INTEGER (1..65535)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Index into the array of sub devices supported."
    ::= { xfsVDMSubDeviceEntry 2 }

-- As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
-- xfsVDMSubDeviceValue    OBJECT-TYPE
-- SYNTAX      Integer32
-- ACCESS      read-only
-- STATUS      current
-- DESCRIPTION  "Returns the value of the sub device referenced by the index."
-- ::= { xfsVDMSubDeviceEntry 3 }
-- *****
-- VDM Error Table
-- *****
-- 1.3.6.1.4.1.16213.2.9.1.4
xfsVDMErrorTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsVDMErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the VDM Error Table."
    ::= { xfsVDMV1 4 }

-- 1.3.6.1.4.1.16213.2.9.1.4.1
xfsVDMErrorEntry OBJECT-TYPE
    SYNTAX XfsVDMErrorEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "VDM Error Table Entry."
    INDEX { xfsVDMErrorManagedServiceName, xfsVDMErrorCommandCode,
xfsVDMErrorResponseCode }
    ::= { xfsVDMErrorTable 1 }

XfsVDMErrorEntry ::=
    SEQUENCE {
        xfsVDMErrorManagedServiceName
            DisplayString,
        xfsVDMErrorCommandCode
            INTEGER,
        xfsVDMErrorResponseCode
            INTEGER,
        xfsVDMErrorCount
            Integer32
    }

-- 1.3.6.1.4.1.16213.2.9.1.4.1.1
xfsVDMErrorManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current

```

```

DESCRIPTION
    "Instance identifier of the managed service."
    ::= { xfsVDMErrorEntry 1 }

-- 1.3.6.1.4.1.16213.2.9.1.4.1.2
xfsVDMErrorCommandCode OBJECT-TYPE
    SYNTAX INTEGER (901..1000)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The executable command code supported by the Service
        Provider associated with the error count of interest."
    ::= { xfsVDMErrorEntry 2 }

-- 1.3.6.1.4.1.16213.2.9.1.4.1.3
xfsVDMErrorResponseCode OBJECT-TYPE
    SYNTAX INTEGER (0..99 | 900..999)
    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."
    ::= { xfsVDMErrorEntry 3 }

-- 1.3.6.1.4.1.16213.2.9.1.4.1.4
xfsVDMErrorCount OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "The counter value corresponding to the managed service,
        command code and response code."
    ::= { xfsVDMErrorEntry 4 }

-- *****
-- VDM Reset Table
-- *****
-- 1.3.6.1.4.1.16213.2.9.1.5
xfsVDMResetTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsVDMResetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Defines the set of MIB Variables for the VDM Reset Table."
    ::= { xfsVDMV1 5 }

-- 1.3.6.1.4.1.16213.2.9.1.5.1
xfsVDMResetEntry OBJECT-TYPE
    SYNTAX XfsVDMResetEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "VDM Reset Table Entry."
    INDEX { xfsVDMResetManagedServiceName }
    ::= { xfsVDMResetTable 1 }

XfsVDMResetEntry ::=
    SEQUENCE {
        xfsVDMResetManagedServiceName
            DisplayString,
        xfsVDMResetAll
            Integer32,
        xfsVDMResetTimestamp
            DisplayString
    }

```

```

    }

-- 1.3.6.1.4.1.16213.2.9.1.5.1.1
xfsVDMResetManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Instance identifier of the managed service."
    ::= { xfsVDMResetEntry 1 }

-- 1.3.6.1.4.1.16213.2.9.1.5.1.2
xfsVDMResetAll 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."
    ::= { xfsVDMResetEntry 2 }

-- 1.3.6.1.4.1.16213.2.9.1.5.1.3
xfsVDMResetTimestamp OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Date and time the last reset of the counters was
        performed."
    ::= { xfsVDMResetEntry 3 }

-- *****
-- VDM Device Capabilities Table
-- *****
-- 1.3.6.1.4.1.16213.2.9.1.7
xfsVDMCapabilitiesTable OBJECT-TYPE
    SYNTAX SEQUENCE OF XfsVDMCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Define the set of MIB Variables for the VDM Capabilities table."
    ::= { xfsVDMV1 7 }

-- 1.3.6.1.4.1.16213.2.9.1.7.1
xfsVDMCapabilitiesEntry OBJECT-TYPE
    SYNTAX XfsVDMCapabilitiesEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "VDM Device Capabilities Table Entry."
    INDEX { xfsVDMCapabilitiesManagedServiceName }
    ::= { xfsVDMCapabilitiesTable 1 }

XfsVDMCapabilitiesEntry ::=
    SEQUENCE {
        xfsVDMCapabilitiesManagedServiceName
            DisplayString,
        xfsVDMCapabilitiesExtraCapability
            OCTET STRING
    }

-- 1.3.6.1.4.1.16213.2.9.1.7.1.1
xfsVDMCapabilitiesManagedServiceName OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current

```

```

DESCRIPTION
    "Instance identifier of the managed service."
    ::= { xfsVDMCapabilitiesEntry 1 }

-- 1.3.6.1.4.1.16213.2.9.1.7.1.100
xfsVDMCapabilitiesExtraCapability OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Vendor dependent additional device capabilities information."
    ::= { xfsVDMCapabilitiesEntry 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
-- *****
-- 1.3.6.1.4.1.16213.3.0.109
xfsVDMDetailedDSCTrap NOTIFICATION-TYPE
    OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
xfsCommonTrapManagedServiceClass, xfsCommonTrapManagedServiceClassName,
xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion, xfsCommonTrapEvent,
        xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsVDMStatusDevice,
xfsVDMStatusNumberSubDevices, xfsVDMStatusService,
        xfsVDMStatusExtraStatus }
    STATUS current
    DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
    ::= { xfsTrapV2 109 }

END

--
-- SMIV2_xfsVDM.mib
--

```

5. Appendix B - C-Header files

5.1 XFSMIBVDM.H

```

/*****
*
* xfsmibvdm.h          CEN/XFS - MIB VDM
*
*                      Version 3.30  --  Feb 6, 2017
*
*****/

#ifndef __inc_xfsmibVDM__h
#define __inc_xfsmibVDM__h

#ifdef __cplusplus
extern "C" {
#endif

enum IxfsVDMStatusService
{
    xfsVDMStatusServiceEnterPending    = 1,
    xfsVDMStatusServiceActive,
    xfsVDMStatusServiceExitPending,
    xfsVDMStatusServiceInactive
} xfsVDMStatusService;

/*****
*
*   MIB Variables for the Status Table
*
*****/
#define xfsVDMStatusManagedServiceName    (1)
#define xfsVDMStatusNumberSubDevices      (2)
#define xfsVDMStatusDevice                (3)
#define xfsVDMStatusService               (4)

#define xfsVDMStatusExtraStatus            (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_xfsmibVDM__h */

```