CEN

WORKSHOP

AGREEMENT

CWA 15748-44

September 2011

ICS 35.240.40

English version

Extensions for Financial Services (XFS) interface specification - Release 3.10 - Part 44: XFS MIB Application Management MIB 3.10

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

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

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

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

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

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

Management Centre: Avenue Marnix 17, B-1000 Brussels

Table of Contents

FOREWORD 3

1. INTRODUCTION 6

2. APPLICATION MANAGEMENT 7

2.1 APPLICATION MANAGEMENT DETAILS 7

2.2 APPLICATION MANAGEMENT TRAPS 8

2.2.1 Managed Application Status Change 8

2.2.2 Application agent heartbeat 9

3. APPENDIX A - APPLICATION MANAGEMENT MIB SUB-TREE 11

3.1 APPLICATION MANAGEMENT MIB IN SMIV2 AND SMIV1 ASN-1 FORMAT 11

4. APPENDIX B - C-HEADER FILES 14

4.1 XFSMIBAPPLICATION.H 14

Foreword

This CWA is revision 3.10 of the XFS interface specification.

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties on 2007-11-29, 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 3.10.

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 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 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 Version 3.10

Part 30: XFS MIB Device Specific Definitions - Printer Device Class MIB 3.10

Part 31: XFS MIB Device Specific Definitions - Identification Card Device Class MIB 3.10

Part 32: XFS MIB Device Specific Definitions - Cash Dispenser Device Class MIB 3.10

Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class MIB 3.10

Part 34: XFS MIB Device Specific Definitions - Check Reader/Scanner Device Class MIB 3.10

Part 35: XFS MIB Device Specific Definitions - Depository Device Class MIB 3.10

Part 36: XFS MIB Device Specific Definitions - Text Terminal Unit Device Class MIB 3.10

Part 37: XFS MIB Device Specific Definitions - Sensors and Indicators Unit Device Class MIB 3.10

Part 38: XFS MIB Device Specific Definitions - Camera Device Class MIB 3.10

Part 39: XFS MIB Device Specific Definitions - Alarm Device Class MIB 3.10

Part 40: XFS MIB Device Specific Definitions - Card Embossing Unit Device Class MIB 3.10

Part 41: XFS MIB Device Specific Definitions - Cash-In Module Device Class MIB 3.10

Part 42: Reserved for future use.

Part 43: XFS MIB Device Specific Definitions - Vendor Dependent Mode Class MIB 3.10

Part 44: XFS MIB Application Management MIB 3.10

Part 45: XFS MIB Device Specific Definitions - Card Dispenser Device Class MIB 3.10

Part 46: XFS MIB Device Specific Definitions - Barcode Reader Device Class MIB 3.10

Part 47: XFS MIB Device Specific Definitions - Item Processing Module Device Class MIB 3.10

Parts 48 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 62: Printer and Scanning Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.02 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.03 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.01 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.0 (see CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.02 (see CWA 14050) to Version 3.10 (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/pages/default.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 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 2010-06-17 and was successfully closed on 2010-12-22.The final text of this CWA was submitted to CEN for publication on 2011-01-27.

This CEN Workshop Agreement is publicly available as a reference document from the National Members of CEN: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and 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 the Application Management MIB to coincide with the release of MIB 1.1.
3.10	December 14, 2010	Update of the MIB to align with XFS 3.10.

1. Introduction

This document provides the specific MIB definition (Management Information Base) variables for the Application Management sub-tree version one, as foreseen by the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document.

The xfsAppMIB version one sub-tree is identified by:

xfsMIBRoot

- xfsManagedApp (1000)
  - xfsAppMIBV1 (1)

The xfsAppMIBV1 sub-tree contains the following variables:

- *xfsConsumerApplication* is the state of the consumer application functionality.
- *xfsSupervisorApplication* is the state of the supervisor functionality.
- *xfsConsumerAppCommStatus* is the status of the communication between the consumer application and the host.
- *xfsExtension* is a list of vendor dependent additional application state information.

The *xfsConsumerApplication* and *xfsSupervisorApplication* variables allow the applications view of the overall state of the terminal to be determined and reported. These variables identify system issues, e.g. the consumer application may be offline while all devices are available - the terminal is then not able to offer transaction services (even if the individual states of all the terminal's devices are online and functioning properly).

The *xfsConsumerAppCommStatus* reports the communication status of the customer application with the host, whether it is online, offline or the communication status is unknown.

This document describes the OID structure for reporting the application state. This MIB reflects the status of the consumer application, the status of the supervisor application and the status of the consumer application communications. The status of XFS devices is separately defined and reported by the XFS device class MIBs.

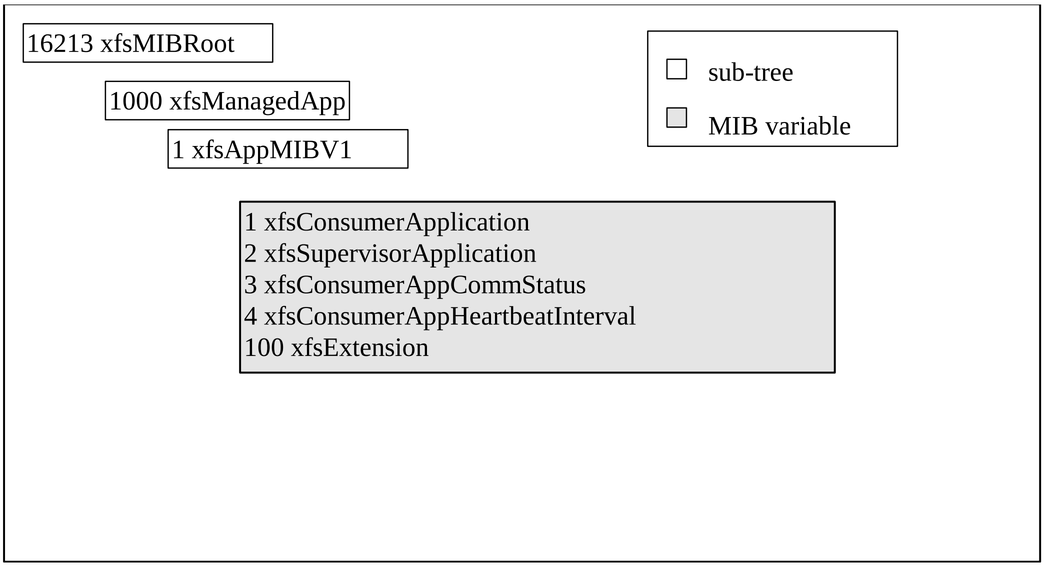
It is important to be clear that this document provides a standard interface for management clients to obtain state information. It does not define an interface for how this information is obtained locally. How this information is populated by local self-service SNMP agents is the responsibility of the agent implementation. In addition, the application management agent implementation must be compatible with the device agent so that the agents can coexist on the same platform. This approach:

- Allows the SNMP agent supplier to define how best that this information be populated.
- Avoids the CEN XFS device standard from encroaching in non-device functionality.

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 *xfsAppMIBV1* sub-tree.

2. Application Management

The following is a graphical representation of the new OID Structure:



2.1 Application Management Details

2.1.1.1 XFS Application Status

xfsMIBRoot	16213
xfsManagedApp	1000
xfsAppMIBV1	1

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerApplication (1)

Specifies the state of the consumer application functionality. It reports the current state of the consumer application as one of the following flags:

Value	Meaning
xfsConsumerAppInService (1)	The application is available and able to offer service (transactions) to the consumer.
xfsConsumerAppInUse (2)	The application is currently in use by a consumer (i.e. a consumer is executing a transaction of some sort).
xfsConsumerAppOutOfService (3)	The application is out of service and is therefore not available to offer services to the consumer.
xfsConsumerAppError (4)	The application is not available to offer services to the consumer because of a fault condition.
xfsConsumerAppUnknown (5)	The agent is unable to report the consumer application state. This may be because the consumer application has not informed it of its state.

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsSupervisorApplication (2)

Specifies the state of the supervisor functionality. It reports the current state of the supervisor application as one of the following flags:

Value	Meaning
xfsSupervisorAppAvailable (1)	The supervisor application is available for selection (i.e. monitoring for requests to go into supervisor mode), but is not currently being used.
xfsSupervisorAppActive (2)	The supervisor application is active and being used currently (i.e. a request to enter supervisor was recognized, and the supervisor application is offering supervisor functions).

xfsSupervisorAppNotAvailable (3)	The supervisor application is not available for selection. Any attempt to request supervisor will fail.
xfsSupervisorAppUnknown (4)	The agent is unable to report the supervisor application state. This may be because the supervisor application has not informed it of its state.

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerAppCommStatus (3)
Specifies the status of the communication between the consumer application and the host. The communication status is reported as one of the following flags:

Value	Meaning
xfsConsumerAppCommOnline (1)	The consumer application is online with the host.
xfsConsumerAppCommOffline (2)	The consumer application is correctly loaded and running but is offline with the host.
xfsConsumerAppCommError (3)	The consumer application is offline with the host because of an application fault condition.
xfsConsumerAppCommUnknown (4)	The consumer application communication status with the host is unknown.

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerAppHeartbeatInterval (4)
Specifies the interval of the heartbeat in minutes. This is an integer read-write variable. A 0 (zero) value will turn the heartbeat off.

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsExtension (100)
Specifies a list of vendor dependent additional application state information. The information is returned as a series of "*key=value"* strings. Each string will be null-terminated, with the final string terminating with two null characters.

2.2 Application Management Traps

The following section defines XFS traps that are specific to application management.

2.2.1 Managed Application Status Change

Status changes within an application should be reported to the SNMP agent. The following section explicitly defines the format of the Managed Application Status Change trap.

The Managed Application Status Change trap should only be generated when the status of the consumer application, supervisor application, or application extension status changes. In addition, this trap is only generated on version 1.1 of the MIB and higher.

The SNMP Specific trap value 1000 defines the trap as an application management trap.

2.2.1.1 Application Management Status Change Trap Format

The following defines the variable bindings included in the Application Management Status Change Trap. The numbers following each OID represent the index of the OID in the trap and not the OID value.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate (1)
This variable represents the UTC and bias for local translation of the date and time when the application trap 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.xfsManagedApp.xfsAppMIBV1.xfsConsumerApplication (2)
This variable binding represents the consumer application state. It is a 32 bit integer field (INT32).

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsSupervisorApplication (3)
This variable binding represents the supervisor application state. It is a 32 bit integer field (INT32).

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerAppCommStatus (4)
This variable binding contains the communication status of the consumer application with the host. It is a 32 bit integer field (INT32).

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsExtension (5)
This variable binding contains vendor dependent additional application state 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 terminated with two null characters.

2.2.1.2 XFS Managed Application Status Change Trap: an example

As an example, the following variable binding list represents a managed application status change trap (1000, 1).

xfsMIBRoot.3.1.3.11	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate) "20/03/2006 15:40:53 -300"
xfsMIBRoot.1000.1.1	(xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerApplication) 1 (xfsConsumerAppInService)
xfsMIBRoot.1000.1.2	(xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsSupervisorApplication) 1 (xfsSupervisorAppAvailable)
xfsMIBRoot.1000.1.3	(xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerAppCommStatus) 1 (xfsConsumerAppCommOnline)
xfsMIBRoot.1000.1.100	(xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsExtension) "key1=value1"

2.2.2 Application agent heartbeat

The application agent heartbeat trap should be generated as per a configurable timer. In addition, this trap is only generated on version 1.1 of the MIB and higher.

The SNMP Specific trap value 1001 defines the trap as an application agent heartbeat trap.

2.2.2.1 Application Agent Heartbeat Trap Format

The following defines the variable bindings included in the application agent heartbeat trap. The numbers following each OID represent the index of the OID in the trap and not the OID value.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName (1)
This variable binding contains the system generating the trap; it is a Display String field.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion (2)
This variable binding represents the XFS MIB version, and 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 (3)
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.xfsHeartbeatTrapVars.xfsHeartbeatTrapMode (4)
This variable binding contains the system state for the Vendor Dependent Mode in integer format. This state provides indication as to whether device states or traps are to be expected. Possible values are:

Value	Meaning
xfsVDMEnterPending(1)	Vendor Dependent Mode enter request pending.
xfsVDMActive(2)	Vendor Dependent Mode active.
xfsVDMExitPending(3)	Vendor Dependent Mode exit request pending.
xfsVDMInactive(4)	Vendor Dependent Mode inactive.

xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerAppHeartbeatInterval (5)
This variable binding contains the heartbeat interval in integer read-write variable. The interval provides an indication as to when the next heartbeat trap should be expected. The value is the number of minutes.

2.2.2.2 Application agent heartbeat Trap: an example

As an example, the following variable binding list represents an application agent heartbeat trap (6, 1001) in a configurable interval of 5 minutes while Vendor Dependent Mode inactive.

xfsMIBRoot.3.1.3.1	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName)
	“SST System 1”
xfsMIBRoot.3.1.3.9	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion)
	“1.20”
xfsMIBRoot.3.1.3.11	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate)
	“15/05/2007 15:40:53 -300”
xfsMIBRoot.3.1.5.1	(xfsMIBRoot.xfsTrap.xfsTrapV1.xfsHeartbeatTrapVars.xfsHeartbeatTrapMode)
	4
xfsMIBRoot.1000.1.4	(xfsMIBRoot.xfsManagedApp.xfsAppMIBV1.xfsConsumerAppHeartbeatInterval)
	5

3. Appendix A - Application Management MIB sub-tree

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

3.1 Application Management MIB in SMIv2 and SMIv1 ASN-1 format

SMIv1_xfsApplication.mib SMIv2_xfsApplication.mib

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

```
--
-- This MIB module has been converted from SMIv1 to SMIv2 format by
-- MG-SOFT Visual MIB Builder.
--
-- Caution: There are differences in values and use of SMIv1 and SMIv2
-- STATUS and ACCESS/MAX-ACCESS clauses. When a MIB module is mechanically
-- translated from SMIv1 to SMIv2 format, the STATUS and ACCESS clause
-- values cannot always be appropriately converted.
--
-- * The SMIv1 STATUS clause values specify two attributes of object
--   definitions: the validity of the object definition and the implementation
--   conformance requirements. The SMIv2 STATUS clause values specify only
--   the validity of the object definition, while the conformance requirements
--   can be defined with the MODULE-COMPLIANCE statements.
--
-- * In SMIv1, the ACCESS clause specifies the minimal required access,
--   while in SMIv2, the MAX-ACCESS clause specifies the maximum allowed access.
--   This should be considered when converting an ACCESS clause to a MAX-ACCESS
clause.
--
-- You should carefully check all STATUS and MAX-ACCESS clause values of OBJECT-
TYPE
-- definitions in this module and modify them or add a note into DECRIPTION clause
-- if necessary. For details about the rules for converting MIB modules from SMIv1
to
-- SMIv2 format, see the RFC 3584 (section 2.1.1), or "Understanding SNMP MIBs"
-- book by D.T. Perkins and E. McGinnis (section 8.2).
--

       XFS-APP-MAN-MIB-V2 DEFINITIONS ::= BEGIN

             IMPORTS
                    Integer32, OBJECT-TYPE, NOTIFICATION-TYPE
                           FROM SNMPv2-SMI
                    xfsMIBRoot, xfsTrap
                           FROM XFSMIB;



--
-- Type definitions
--

--  *********************************************************************
-- General #defines
-- *********************************************************************
             IxfsConsumerAppStatus ::= INTEGER
                    {
                    xfsInService(1),
                    xfsInUse(2),
                    xfsOutOfService(3),
                    xfsUnknown(4)
                    }

             IxfsSupervisorAppStatus ::= INTEGER
                    {
```

```
                    xfsSuperAvailable(1),
                    xfsSuperActive(2),
                    xfsSuperNotAvailable(3),
                    xfsSuperUnknown(4)
                    }

             IxfsConsumerAppCommStatus ::= INTEGER
                    {
                    xfsConsumerAppCommOnline(1),
                    xfsConsumerAppCommOffline(2),
                    xfsConsumerAppCommError(3),
                    xfsConsumerAppCommUnknown(4)
                    }


--
-- Node definitions
--

             -- 1.3.6.1.4.1.16213.3.0
             xfsTrapV2 OBJECT IDENTIFIER ::= { xfsTrap 0 }


--  **********************************************************************
-- Managed Application Status Change Trap
-- **********************************************************************
             -- 1.3.6.1.4.1.16213.3.0.1000
             xfsManagedAppTrap NOTIFICATION-TYPE
                    OBJECTS { xfsCommonTrapDate, xfsConsumerApplication,
xfsSupervisorApplication, xfsExtension }
                    STATUS current
                    DESCRIPTION
                           "This trap indicates a change in application status"
                    ::= { xfsTrapV2 1000 }


--  **********************************************************************
-- Application agent heartbeat Trap
-- **********************************************************************
             -- 1.3.6.1.4.1.16213.3.0.1001
             xfsManagedAppHeartbeatTrap NOTIFICATION-TYPE
                    OBJECTS { xfsCommonTrapSysName, xfsCommonTrapMIBVersion,
xfsCommonTrapDate, xfsHeartbeatTrapMode, xfsConsumerAppHeartbeatInterval }
                    STATUS current
                    DESCRIPTION
                           "This trap indicates that the application SNMP agent is
up and available"
                    ::= { xfsTrapV2 1001 }


             -- 1.3.6.1.4.1.16213.1000
             xfsManagedApp OBJECT IDENTIFIER ::= { xfsMIBRoot 1000 }


             -- 1.3.6.1.4.1.16213.1000.1
             xfsAppMIBV1 OBJECT IDENTIFIER ::= { xfsManagedApp 1 }


             -- 1.3.6.1.4.1.16213.1000.1.1
             xfsConsumerApplication OBJECT-TYPE
                    SYNTAX IxfsConsumerAppStatus
                    MAX-ACCESS read-only
                    STATUS current
                    DESCRIPTION
                           "Consumer Application Status"
                    ::= { xfsAppMIBV1 1 }


             -- 1.3.6.1.4.1.16213.1000.1.2
             xfsSupervisorApplication OBJECT-TYPE
                    SYNTAX IxfsSupervisorAppStatus
                    MAX-ACCESS read-only
                    STATUS current
                    DESCRIPTION
                          "Supervisor Application Status"
                    ::= { xfsAppMIBV1 2 }


             -- 1.3.6.1.4.1.16213.1000.1.3
             xfsConsumerAppCommStatus OBJECT-TYPE
                    SYNTAX IxfsConsumerAppCommStatus
                    MAX-ACCESS read-only
                    STATUS current
                    DESCRIPTION
                          "Status of the communication between the consumer
application and the host"
                    ::= { xfsAppMIBV1 3 }


             -- 1.3.6.1.4.1.16213.1000.1.4
             xfsConsumerAppHeartbeatInterval OBJECT-TYPE
                    SYNTAX Integer32
                    MAX-ACCESS read-write
                    STATUS current
                    DESCRIPTION
                          "Interval of the heartbeat in minutes"
                    ::= { xfsAppMIBV1 4 }


             -- 1.3.6.1.4.1.16213.1000.1.100
             xfsExtension OBJECT-TYPE
                    SYNTAX OCTET STRING
                    MAX-ACCESS read-only
                    STATUS current
                    DESCRIPTION
                          "Vendor dependent additional application state
information"
                    ::= { xfsAppMIBV1 100 }



       END

--
--
--
```

4. Appendix B - C-Header files

4.1 XFSMIBApplication.H

```
/************************************************************************
*                                                                       *
* xfsmibapplication.h   CEN/XFS - MIB Application Management            *
*                                                                       *
*               Version 3.10  --  Dec 14, 2010                          *
*                                                                       *
************************************************************************/

#ifndef __inc_xfsmibapplication__h
#define __inc_xfsmibapplication__h

#ifdef __cplusplus
extern "C" {
#endif

enum IxfsConsumerAppStatus
{
    xfsConsumerAppInService         = 1,
    xfsConsumerAppInUse,
    xfsConsumerAppOutOfService,
    xfsConsumerAppError,
    xfsConsumerAppUnknown

} xfsConsumerAppStatus;

enum IxfsSuppervisorAppStatus
{
    xfsSupervisorAppAvailable       = 1,
    xfsSupervisorAppActive,
    xfsSupervisorAppNotAvailable,
    xfsSupervisorAppUnknown

} xfsSuppervisorAppStatus;

enum IxfsConsumerAppCommStatus
{
    xfsConsumerAppCommOnline        = 1,
    xfsConsumerAppCommOffline,
    xfsConsumerAppCommError,
    xfsConsumerAppCommUnknown

} xfsConsumerAppCommStatus;

/************************************************************************
*                                                                       *
*    MIB Variables for the Application Management Table                 *
*                                                                       *
************************************************************************/
#define xfsConsumerApplication              (1)
#define xfsSupervisorApplication            (2)
#define xfsConsumerAppCommStatus            (3)
#define xfsConsumerAppHeartbeatInterval     (4)
#define xfsExtension                        (100)

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