CEN

WORKSHOP

AGREEMENT

CWA 14050-44

July 2007

ICS 35.200; 35.240.15; 35.240.40

English version

Extensions for Financial Services (XFS) interface specification - Release 3.03 - Part 44: XFS MIB Application Management MIB 1.1

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

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

Ref. No.:CWA 14050-44:2007 D/E/F

Table of Contents

Foreword

This CWA is revision 3.03 of the XFS interface specification.

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

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

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

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

Part 2: Service Classes Definition; Programmer's Reference

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Part 44: XFS MIB Application Management

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

Revision History:

1,1	15 April 2007	Initial release of the Application Management MIB to coincide with XFS MIB release 1.1

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

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

1. Introduction

This document provides the 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.

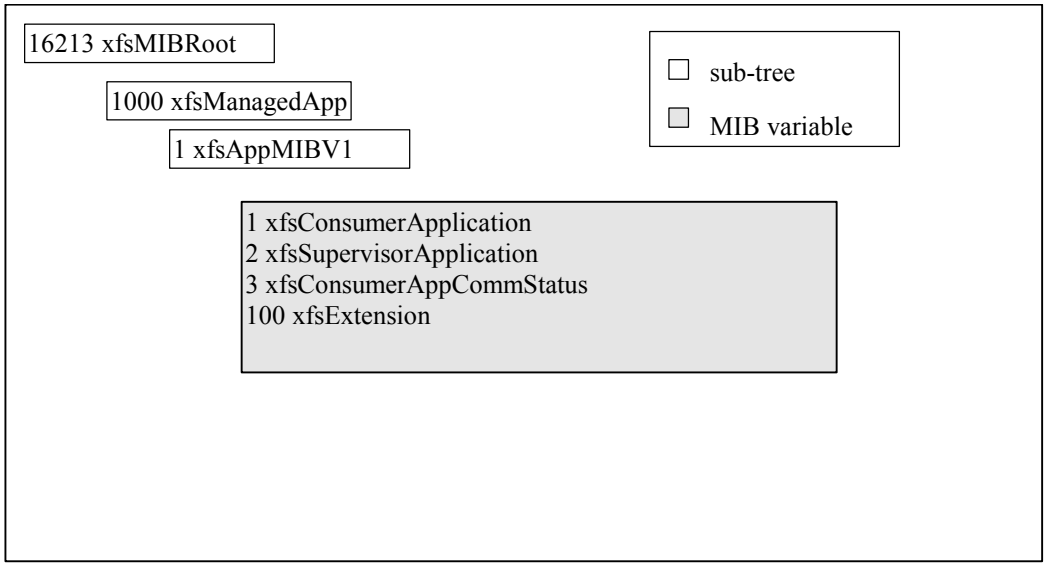
It is important to be clear that this document provides a standard interface for management clients to obtain state information. It does not define an interface for how this information is obtained locally. How this information is populated by local self-service SNMP agents is the responsibility of the 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.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 32bit integer field.

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"

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

The following object contains the xfsApplication.MIB file in SMIv2 format.

SMIv2\
xfsApplication.mib

The following object contains the xfsApplication.MIB file in SMIv2 format.

SMIv1\
xfsApplication.mib

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

```
-- ****************************************************************
-- XFS MIB for Application Management
-- Management Information Base for XFS Application Management
-- ****************************************************************
 XFS-APP-MAN-MIB DEFINITIONS ::= BEGIN

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


-- ****************************************************************
-- General #defines
-- ****************************************************************
   IxfsConsumerAppStatus ::= INTEGER
       {
       xfsConsumerAppInService(1),
       xfsConsumerAppInUse(2),
       xfsConsumerAppOutOfService(3),
       xfsConsumerAppError(4),
       xfsConsumerAppUnknown(5)
       }

   IxfsSupervisorAppStatus ::= INTEGER
       {
       xfsSupervisorAppAvailable(1),
       xfsSupervisorAppActive(2),
       xfsSupervisorAppNotAvailable(3),
       xfsSupervisorAppUnknown(4)
       }

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

   xfsManagedApp     OBJECT IDENTIFIER ::= {xfsMIBRoot 1000}
   xfsAppMIBV1       OBJECT IDENTIFIER ::= {xfsManagedApp 1}
```

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

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

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

   xfsExtension OBJECT-TYPE
       SYNTAX       OCTET STRING
       MAX-ACCESS   read-only
       STATUS       current
       DESCRIPTION  ""
       ::= {xfsAppMIBV1 100}

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

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

END
```

4. Appendix B - C-Header files

4.1 XFSMIBApplication.H

xfsMIBApplication.h

```
/******************************************************************************
*                                                                             *
* xfsmibapplication.h   WOSA/XFS - MIB Application Management                 *
*                                                                             *
*                       Version 1.00  --  Apr 14, 2006                        *
*                                                                             *
******************************************************************************/

#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 xfsExtension                    (100)

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