# CEN

# WORKSHOP

# AGREEMENT

# CWA 15748-33

September 2011

**ICS** 35.240.40

English version

# Extensions for Financial Services (XFS) interface specification - Release 3.10 - Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class 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**



Ref. No.:CWA 15748-33:2011 D/E/F

# Table of Contents

# 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:

| | | |
|---|---|---|
| 1.0 | January 20, 2004 | Initial release of XFS MIB specification. |
| 1.10 | April 15, 2007 | Update of the MIB to add support for a Detailed Status Trap, a Device Reset capability and the support of SMIv2. |
| 3.10 | December 14, 2010 | Update of the MIB to add support for a Capabilities table and to align the MIB with XFS 3.10. |

# 1 Introduction

This document provides the device specific MIB definition (Management Information Base) variables for the xfsPIN sub-tree version one, as foreseen by the *XFS MIB Architecture and SNMP Extensions Programmer's reference* document. All the attributes in all the MIBs are Mandatory. In the case where a vendor's device does not support an attribute then a request for this unsupported attribute should return NULL.

The xfsPIN version one sub-tree is identified by:

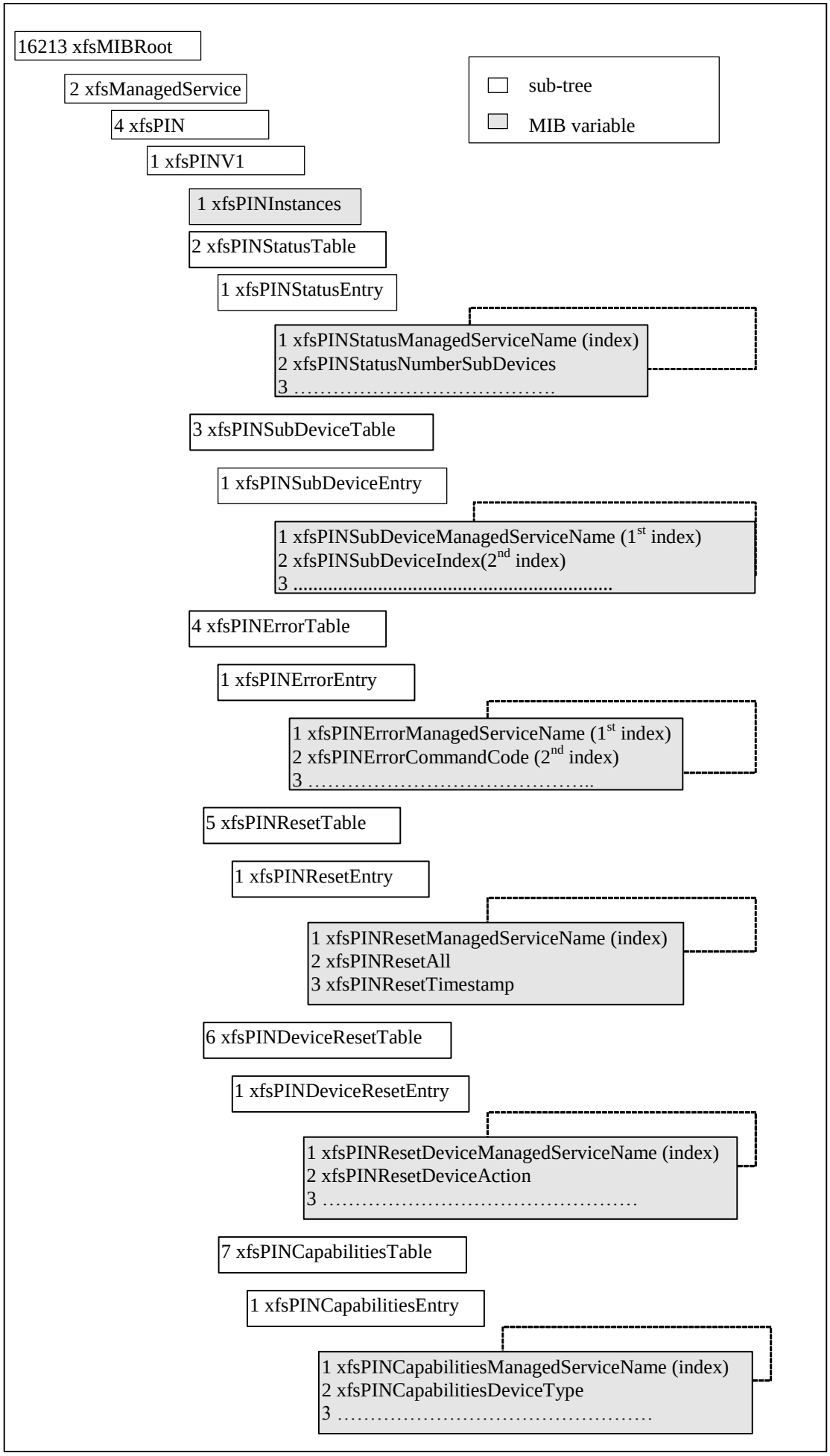
xfsMIBRoot

- xfsManagedService (2)
  - xfsPIN (4)
    - xfsPINV1 (1)

The xfsPINV1 sub-tree contains the following variables:

* *xfsPINInstances(1)* is the number of managed services for the PIN class installed on the XFS subsystem. It is a 32 bit numerical field.
* *xfsPINStatusTable(2)* identifies the table for the PIN variables.
* *xfsPINSubDeviceTable(3)* not applicable to the PIN device.
* *xfsPINErrorTable(4)* identifies the table for the PIN error counters.
* *xfsPINResetTable(5)* identifies the table for the PIN reset variable.
* *xfsPINResetDeviceTable(6)* identifies the table for the PIN reset device variables.
* *xfsPINCapabilitiesTable(7)* identifies the table for the PIN 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 *xfsPINV1* sub-tree.


16213 xfsMIBRoot
2 xfsManagedService
4 xfsPIN
1 xfsPINV1
1 xfsPINInstances
2 xfsPINStatusTable
1 xfsPINStatusEntry
1 xfsPINStatusManagedServiceName (index)
2 xfsPINStatusNumberSubDevices
3 ……
3 xfsPINSubDeviceTable
1 xfsPINSubDeviceEntry
1 xfsPINSubDeviceManagedServiceName (1st index)
2 xfsPINSubDeviceIndex(2nd index)
3 .....
4 xfsPINErrorTable
1 xfsPINErrorEntry
1 xfsPINErrorManagedServiceName (1st index)
2 xfsPINErrorCommandCode (2nd index)
3 ……..
5 xfsPINResetTable
1 xfsPINResetEntry
1 xfsPINResetManagedServiceName (index)
2 xfsPINResetAll
3 xfsPINResetTimestamp
6 xfsPINDeviceResetTable
1 xfsPINDeviceResetEntry
1 xfsPINResetDeviceManagedServiceName (index)
2 xfsPINResetDeviceAction
3 ……
7 xfsPINCapabilitiesTable
1 xfsPINCapabilitiesEntry
1 xfsPINCapabilitiesManagedServiceName (index)
2 xfsPINCapabilitiesDeviceType
3 ……
sub-tree
MIB variable

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

# 2 XFS PIN MIB variables

This section describes the MIB variables for the tables of the PIN Class. The description of the variables listed below includes, where it is meaningful, a reference to relevant data structures and commands defined inside the *Pin Keypad Device Class Interface Programmer's Reference.*
Note that all counters persist across re-boot. The following are some general notes pertaining to the MIB variables:

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

## 2.1 XFS PIN Status Table

The *xfsPINStatusTable(2)* groups the variables identifying device status information, statistics and auxiliary variables. It is indexed through a single parameter, *xfsPINStatusManagedServiceName*. 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.

*xfsPINStatusManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the PIN class.

As an example, the identifier for the device status value of *xfsPINStatusEncStat(4)* for a device with managed service name equal to "PinPad1" is as follows:

| Character | P | i | n | P | a | d | 1 |
|---|---|---|---|---|---|---|---|
| ASCII Hex | 50 | 69 | 6E | 50 | 61 | 64 | 31 |
| ASCII Dec | 80 | 105 | 110 | 80 | 97 | 100 | 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.2.4.1.2.1.4.7.80.105.110.80.97.100.49*

### 2.1.1 xfsPINStatusTable: States

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

xfsPINStatusManagedServiceName (1)
Uniquely identifies the managed service.

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

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

| Value | Meaning |
|---|---|
| xfsDevOnline(1) | The device is online (i.e., powered on and operable). |
| xfsDevOffline(2) | The device is offline (e.g., the operator has taken the device offline by turning a switch or pulling out the device). |

| | |
|---|---|
| xfsDevPowerOff(3) | The device is powered off or physically not connected. |
| xfsDevNoDevice(4) | There is no device intended to be there; e.g. this type of self service machine does not contain such a device or it is internally not configured. |
| xfsDevHWError(5) | The device is inoperable due to a hardware error. |
| xfsDevUserError(6) | The device is inoperable because a person is preventing proper device operation. |
| xfsDevBusy(7) | The device is busy and unable to process an execute command at this time. |
| xfsDevFraudAttempt(8) | The device is present but has detected a fraud attempt. |

xfsPINStatusEncStat (4)

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

| Value | Meaning |
|---|---|
| xfsPINEncReady (1) | The encryption module is initialized and ready (at least one key is imported into the encryption module). |
| xfsPINEncNotReady (2) | The encryption module is not available or not ready due to hardware error or communication error. |
| xfsPINEncNotInitialized (3) | The encryption module is not initialized (no master key loaded). |
| xfsPINEncBusy (4) | The encryption module is busy (implies that the device is busy). |
| xfsPINEncUndefined (5) | The encryption module state is undefined. |
| xfsPINEncInitialized (6) | The encryption module is initialized and master key (where required) and any other initial keys are loaded; ready to import other keys. |

xfsPINStatusGuidancePinPad (5)

It contains the state of the guidance light on the PIN pad unit. Values are reported as a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000000 | WFS_PIN_GUIDANCE_NOT_AVAILABLE | The status is not available. |
| 0x00000001 | WFS_PIN_GUIDANCE_OFF | The light is turned off. |
| 0x00000004 | WFS_PIN_GUIDANCE_SLOW_FLASH | The light is blinking slowly. |
| 0x00000008 | WFS_PIN_GUIDANCE_MEDIUM_FLASH | The light is blinking medium frequency. |
| 0x00000010 | WFS_PIN_GUIDANCE_QUICK_FLASH | The light is blinking quickly. |
| 0x00000080 | WFS_PIN_GUIDANCE_CONTINUOUS | The light is turned on continuous (steady). |
| 0x00000100 | WFS_PIN_GUIDANCE_RED | The light is red. |
| 0x00000200 | WFS_PIN_GUIDANCE_GREEN | The light is green. |
| 0x00000400 | WFS_PIN_GUIDANCE_YELLOW | The light is yellow. |
| 0x00000800 | WFS_PIN_GUIDANCE_BLUE | The light is blue. |
| 0x00001000 | WFS_PIN_GUIDANCE_CYAN | The light is cyan. |
| 0x00002000 | WFS_PIN_GUIDANCE_MAGENTA | The light is magenta. |
| 0x00004000 | WFS_PIN_GUIDANCE_WHITE | The light is white. |

xfsPINStatusAutoBeepMode (6)

It contains whether automatic beep tone on key press is active or not. It is a numeric type field. Active and in-active key beeping is reported independently. If the flag is not set auto beeping is not activated (or not supported) for that key type (i.e. active or inactive keys). Values are reported as a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_BEEP_ON_ACTIVE | An automatic tone will be generated for all active keys. |
| 0x0002 | WFS_PIN_BEEP_ON_INACTIVE | An automatic tone will be generated for all in-active keys. |

xfsPINStatusCertificateState (7)

It contains the state of the public verification or encryption key in the PIN certificate modules. It is a numeric type field. Allowed values are as follows:

| Value | Meaning |
|---|---|
| xfsPINCertUnknown (1) | The state of the certificate module is unknown or the device does not have this capability. |
| xfsPINCertPrimary (2) | All pre-loaded certificates have been loaded and that primary verification certificates will be accepted for the commands WFS_CMD_PIN_LOAD_CERTIFICATE or WFS_CMD_PIN_REPLACE_CERTIFICATE. |
| xfsPINCertSecondary (3) | Primary verification certificates will not be accepted and only secondary verification certificates will be accepted. If primary certificates have been compromised (which the certificate authority or the host detects), then secondary certificates should be used in any transaction. This is done by calling the WFS_CMD_PIN_LOAD_CERTIFICATE command or the WFS_CMD_PIN_REPLACE_CERTIFICATE. |
| xfsPINCertNotReady (5) | The certificate module is not ready.(The device is powered off or physically not present). |

xfsPINStatusDevicePosition (8)

It contains the device position. It is a numeric type field. Allowed values are:

| Value | Meaning |
|---|---|
| xfsPINDeviceInPosition (1) | The device is in its normal operating position, or is fixed in place and cannot be moved. |
| xfsPINDeviceNotInPosition (2) | The device has been removed from its normal operating position. |
| xfsPINDevicePosUnknown (3) | Due to a hardware error or other condition, the position of the device cannot be determined. |
| xfsPINDevicePosNotSupp (4) | The physical device does not have the capability of detecting the position. |

xfsPINStatusPowerSaveRecoveryTime (9)

It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. This value is zero if either the power saving mode has not been activated or no power save control is supported. The value is a numeric type field.

xfsPINStatusExtraStatus (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 an additional null character. An empty list is indicated by two consecutive null characters.

## 2.2 XFS PIN Sub Device Table

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

The *xfsPINErrorTable(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:

*xfsPINErrorManagedServiceName*
*xfsPINErrorCommandCode*

*xfsPINErrorResponseCode*

The *xfsPINErrorTable* is defined as:

- *xfsPINErrorManagedServiceName(1*) which provides the primary index to the service in question. It is Display String field. The *xfsPINErrorManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table, e.g. "PinPad1".
- *xfsPINErrorCommandCode(2)* is an index which identifies the command code that that response code is related to, e.g. WFS_CMD_PIN_CRYPT (401). It is a 32 bit numerical field.
- *xfsPINErrorResponseCode(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_PIN_KEYNOTFOUND (-400) is represented by 400. It is a 32 bit numerical field.
- *xfsPINErrorCount(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_PIN_KEYNOTFOUND (-400) error returned from the WFS_CMD_PIN_CRYPT (401) command for a device with managed service name equal to "PinPad1" is as follows:

*xfsMIBRoot.2.4.1.4.1.4.7.80.105.110.80.97.100.49.401.400.*

## 2.4 XFS PIN Reset Table

The *xfsPINResetTable(5)* contains the *xfsPINResetAll* and *xfsPINResetTimestamp* variables and is indexed by the single variable, *xfsPINResetManagedServiceName*. When the *xfsPINResetAll* 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 *xfsPINResetTable* is defined as:

- *xfsPINResetManagedServiceName(1)* which provides the index to the service in question. It is Display String field. The *xfsPINResetManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table, e.g. "PinPad1".
- *xfsPINResetAll(2)* is a read-write variable. Issue of a Set command on the *xfsPINResetAll* 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 *xfsPINResetAll* variable will return 0 (zero).
- *xfsPINResetTimestamp(3)* is a read-only variable which represents the UTC date and time when the counters in the error table was reset, it is a Display String field. The data is formatted in the following way: "DD/MM/YYYY HH:MM:SS +ZZZ" where DD/MM/YYYY HH:MM:SS is the local date and time. ZZZ is the bias, which is the difference, in minutes, between Coordinated Universal Time (UTC) and local time.

As an example, all the error counts can be reset for a device with managed service name equal to "PinPad1" by setting the value zero in the *xfsPINResetAll* variable represented by:

*xfsMIBRoot.2.4.1.5.1.2.7.80.105.110.80.97.100.49*

## 2.5 XFS PIN Reset Device Table

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

The *xfsPINResetDeviceAction* variable is used to initiate a reset. Setting this variable will cause the following to happen:

1. The SNMP agent will determine if a Device Reset is allowed by checking the *RemoteDeviceResetAllowed* configuration flag (see XFS Common Management Configuration section, within the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document). If it is not allowed then the flow continues with step 5, otherwise the flow continues with step 2.
2. Exclusive access to the device will be obtained.
3. A WFS_CMD_PIN_RESET command will be issued.
4. Exclusive access to the device will be relinquished when the WFS_CMD_PIN_RESET command completes.

   NOTE Exclusive access must be relinquished as soon as possible and implemented in such a way that deadlocks are avoided.
5. A *xfsPINResetDeviceCompleteTrap* trap will be generated to report the result of the Device Reset request.

The *xfsPINResetDeviceTable* is defined as:

- *xfsPINResetDeviceManagedServiceName(1)* which provides the index to the service in question. It is a Display String field. The *xfsPINResetDeviceManagedServiceName* parameter corresponds to the value of *xfsMIBRoot.xfsGeneral.xfsMIBV1.xfsManagedServiceTable.xfsManagedServiceEntry.xfsManagedServiceName* in the general table, e.g. "PinPad1".
- *xfsPINResetDeviceAction(2)* is a read-write variable. Issue of a Set command on the *xfsPINResetDeviceAction* variable with value *executeReset(1)* will result in the device being reset as described above.
- *xfsPINResetDeviceMediaControl(3)* is a read-only variable. As there is no media in the PIN device class this variable can only report the *mediaDefault* value.
- *xfsPINResetDeviceStatus(4)* is a read only variable This variable can be used to check if a reset operation is still in progress. It is set when the reset is initiated and cleared when the reset command completes.

As an example, the device with managed service name equal to "PinPad1" is reset by setting the *xfsPINResetDeviceAction* variable represented by:

*xfsMIBRoot.2.4.1.6.1.2.7.80.105.110.80.97.100.49*

## 2.6 XFS PIN Capabilities Table

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

*xfsPINCapabilitiesManagedServiceName* is the instance identifier of the managed service and uniquely identifies one instance of the PIN class.

As an example, the identifier for the device status value of *xfsPINCapabilitiesDeviceType(2)* for a device with managed service name equal to "PinPad1" is as follows:

| Character | P | i | n | P | a | d | 1 |
|---|---|---|---|---|---|---|---|
| ASCII Hex | 50 | 69 | 6E | 50 | 61 | 64 | 31 |
| ASCII Dec | 80 | 105 | 110 | 80 | 97 | 100 | 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.2.4.1.7.1.2.7.80.105.110.80.97.100.49*

### 2.6.1 xfsPINCapabilitiesTable: Capabilities

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

xfsPINCapabilitiesManagedServiceName (1)
Uniquely identifies the managed service.

xfsPINCapabilitiesDeviceType (2)
Specifies the type of the PIN pad security module. PIN entry is only possible when at least WFS_PIN_TYPEEPP and WFS_PIN_TYPEEDM are set. In order to use the ZKA-Electronic purse, all flags must be set. It is a numeric variable. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_TYPEEPP | Electronic PIN pad (keyboard data entry device). |
| 0x0002 | WFS_PIN_TYPEEDM | Encryption/decryption module. |
| 0x0004 | WFS_PIN_TYPEHSM | Hardware security module (electronic PIN pad and encryption module within the same physical unit). |

xfsPINCapabilitiesCompoundDevice (3)
Specifies whether the logical device is part of a compound physical device as a TruthValue variable.

| Value | Meaning |
|---|---|
| True(1) | The device is a compound device. |
| False(2) | The device is not a compound device. |

xfsPINCapabilitiesKeyNumber (4)
Specifies the number of the keys which can be stored in the encryption/decryption module as an integer variable.

xfsPINCapabilitiesAlgorithm (5)
Specifies the supported encryption modes; a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_CRYPTDESECB | Electronic Code Book. |
| 0x0002 | WFS_PIN_CRYPTDESCBC | Cipher Block Chaining. |
| 0x0004 | WFS_PIN_CRYPTDESCFB | Cipher Feed Back. |
| 0x0008 | WFS_PIN_CRYPTRSA | RSA Encryption. |
| 0x0010 | WFS_PIN_CRYPTECMA | ECMA Encryption. |
| 0x0020 | WFS_PIN_CRYPTDESMAC | MAC calculation using CBC. |
| 0x0040 | WFS_PIN_CRYPTTRIDESECB | Triple DES with Electronic Code Book. |
| 0x0080 | WFS_PIN_CRYPTTRIDESCBC | Triple DES with Cipher Block Chaining. |
| 0x0100 | WFS_PIN_CRYPTTRIDESCFB | Triple DES with Cipher Feed Back. |
| 0x0200 | WFS_PIN_CRYPTTRIDESMAC | Last Block Triple DES MAC as defined in ISO/IEC 9797-1:1999, using: block length n=64, Padding Method 1 (when bPadding=0 ), MAC Algorithm 3, MAC length m where 32<=m<=64. |
| 0x0400 | WFS_PIN_CRYPTMAAMAC | MAC calculation using the Message authenticator algorithm as defined in ISO 8731-2. |

xfsPINCapabilitiesPinFormats (6)

Specifies the supported pin formats as a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_FORM3624 | PIN left justified, filled with padding characters, PIN length 4-16 digits. The Padding Character is a Hexadecimal Digit in the range 0x00 to 0x0F. |
| 0x0002 | WFS_PIN_FORMANSI | PIN is preceded by 0x00 and the length of the PIN (0x04 to 0x0C), filled with padding character 0x0F to the right, PIN length 4-12 digits, XORed with PAN (Primary Account Number, minimum 12 digits without check number) |
| 0x0004 | WFS_PIN_FORMISO0 | PIN is preceded by 0x00 and the length of the PIN (0x04 to 0x0C), filled with padding character 0x0F to the right, PIN length 4-12 digits, XORed with PAN (Primary Account Number without check number, no minimum length specified, missing digits are filled with 0x00) |
| 0x0008 | WFS_PIN_FORMISO1 | PIN is preceded by 0x01 and the length of the PIN (0x04 to 0x0C), padding characters are taken from a transaction field (10 digits). |
| 0x0010 | WFS_PIN_FORMECI2 | (similar to WFS_PIN_FORM3624), PIN only 4 digits |
| 0x0020 | WFS_PIN_FORMECI3 | PIN is preceded by the length (digit), PIN length 4-6 digits, the padding character can range from X'0' through X'F'. |
| 0x0040 | WFS_PIN_FORMVISA | PIN is preceded by the length (digit), PIN length 4-6 digits. If the PIN length is less than six digits the PIN is filled with X'0' to the length of six, the padding character can range from X ' 0 through X ' 9 (This format is also referred to as VISA2). |
| 0x0080 | WFS_PIN_FORMDIEBOLD | PIN is padded with the padding character and may be not encrypted, single encrypted or double encrypted. |
| 0x0100 | WFS_PIN_FORMDIEBOLDCO | PIN with the length of 4 to 12 digits, each one with a value of X'0' to X'9', is preceded by the one-digit coordination number with a value from X'0' to X'F', padded with the padding character with a value from X'0' to X'F' and may be not encrypted, single encrypted or double encrypted. |
| 0x0200 | WFS_PIN_FORMVISA3 | PIN with the length of 4 to 12 digits, each one with a value of X'0' to X'9', is followed by a delimiter with the value of X'F' and then padded by the padding character with a value between X'0' to X'F'. |
| 0x0400 | WFS_PIN_FORMBANKSYS | PIN is encrypted and formatted according to the Banksys Pin Block specifications. |
| 0x0800 | WFS_PIN_FORMEMV | The PIN block is constructed as follows: PIN is preceded by 0x02 and the length of the PIN (0x04 to 0x0C), filled with padding character 0x0F to the right, formatted up to 248 bytes of other data as defined within the EMV 4.0 specifications and finally encrypted with an RSA |

| | | |
|---|---|---|
| | | key. |
| 0x2000 | WFS_PIN_FORMISO3 | PIN is preceded by 0x03 and the length of the PIN (0x04 to 0x0C), padding characters sequentially or randomly chosen, XORed with digits from PAN. |

xfsPINCapabilitiesDerivationAlgorithms (7)

Specifies the supported derivation algorithms as a combination of the following flags as integer value:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_CHIP_ZKA | Algorithm for the derivation of a chip card individual key as described by the German ZKA. |

xfsPINCapabilitiesPresentationAlgorithms (8)

Specifies the supported presentation algorithms as a combination of the following flags as integer value:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_PRESENT_CLEAR | Algorithm for the presentation of a clear text PIN to a chip card. Each digit of the clear text PIN is inserted as one nibble (= half byte) into *lpbChipData.* |

xfsPINCapabilitiesDisplay (9)

Specifies the type of the display used in the PIN pad module as one of the following flags as integer value.

| Value | Meaning |
|---|---|
| xfsPINDispNone(2) | No display unit. |
| xfsPINDispLedThrough(3) | Lights next to text guide user. |
| xfsPINDispDisplay(4) | A real display is available (this does not apply for self-service). |

xfsPINCapabilitiesConnect (10)

Specifies whether the PIN pad is directly physically connected to the ID card Unit as a TruthValue variable.

| Value | Meaning |
|---|---|
| True(1) | The device is directly physically connected. The PIN will be transported securely during the command WFS_CMD_PIN_PRESENT_IDC. |
| False(2) | The device is not directly physically connected. |

xfsPINCapabilitiesIDKey (11)

Specifies if key owner identification (in commands referenced as *lpxIdent*), which authorizes access to the encryption module, is required. A zero value is returned if the encryption module does not support this capability. Otherwise it will be combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_IDKEYINITIALIZATION | ID key is returned by the WFS_CMD_PIN_INITIALIZATION command. |
| 0x0002 | WFS_PIN_IDKEYIMPORT | ID key is required as input for theWFS_CMD_PIN_IMPORT_KEY and WFS_CMD_PIN_DERIVE_KEY command. |

xfsPINCapabilitiesValidationAlgorithms (12)

Specifies the algorithms for PIN validation supported by the service; combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_DES | DES algorithm. |
| 0x0002 | WFS_PIN_EUROCHEQUE | EUROCHEQUE algorithm. |
| 0x0004 | WFS_PIN_VISA | VISA algorithm. |
| 0x0008 | WFS_PIN_DES_OFFSET | DES offset generation algorithm. |
| 0x0010 | WFS_PIN_BANKSYS | Banksys algorithm. |

xfsPINCapabilitiesKeyCheckModes (13)

Specifies the key check modes that are supported to check the correctness of an imported key value; can be a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0000 | WFS_PIN_KCVNONE | No check value. |
| 0x0001 | WFS_PIN_KCVSELF | The key check value is created by an encryption of the key with itself. For a double length key the KCV is generated using 3DES encryption using the first half of the key as the source data for the encryption. |
| 0x0002 | WFS_PIN_KCVZERO | The key check value is created by a zero value with the key. |

xfsPINCapabilitiesGuidancePinPad (14)

Specifies whether the guidance light indicator on the PIN pad unit is available. States are reported as a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000000 | WFS_PIN_GUIDANCE_NOT_AVAILABLE | Guidance is not available. |
| 0x00000001 | WFS_PIN_GUIDANCE_OFF | Guidance is off. |
| 0x00000004 | WFS_PIN_GUIDANCE_SLOW_FLASH | Guidance state is slow flash. |
| 0x00000008 | WFS_PIN_GUIDANCE_MEDIUM_FLASH | Guidance state is medium flash. |
| 0x00000010 | WFS_PIN_GUIDANCE_QUICK_FLASH | Guidance state is quick flash. |
| 0x00000080 | WFS_PIN_GUIDANCE_CONTINUOUS | Guidance state is continuous. |
| 0x00000100 | WFS_PIN_GUIDANCE_RED | Guidance is red. |
| 0x00000200 | WFS_PIN_GUIDANCE_GREEN | Guidance is green. |
| 0x00000400 | WFS_PIN_GUIDANCE_YELLOW | Guidance is yellow. |
| 0x00000800 | WFS_PIN_GUIDANCE_BLUE | Guidance is blue. |
| 0x00001000 | WFS_PIN_GUIDANCE_CYAN | Guidance is cyan. |
| 0x00002000 | WFS_PIN_GUIDANCE_MAGENTA | Guidance is magenta. |
| 0x00004000 | WFS_PIN_GUIDANCE_WHITE | Guidance is white. |

xfsPINCapabilitiesPINCanPersistAfterUse (15)

Specifies whether the device can retain the PIN after a pin processing command e.g. WFS_CMD_PIN_GET_PINBLOCK, WFS_CMD_PIN_LOCAL_DES, WFS_CMD_PIN_PRESENT_IDC, etc. as a TruthValue variable.

| Value | Meaning |
|---|---|
| True (1) | Applications may request, through the WFS_CMD_PIN_MAINTAIN_PIN command that the PIN continues to be held within the device after use by a PIN processing command. |

| | |
|---|---|
| False (2) | The PIN will always be cleared by the device after processing. The WFS_CMD_PIN_MAINTAIN_PIN is not supported. |

xfsPINCapabilitiesAutoBeep (16)

Specifies whether the PIN device will emit a key beep tone on key presses (of active keys or inactive keys), and if so, which mode it supports. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_BEEP_ACTIVE_AVAILABLE | Automatic beep tone on active key key-press is supported. If this flag is not set then automatic beeping for active keys is not supported. |
| 0x0002 | WFS_PIN_BEEP_ACTIVE_SELECTABLE | Automatic beeping for active keys can be controlled (i.e. turned on and off) by the application. If this flag is not set then automatic beeping for active keys cannot be controlled by an application. |
| 0x0004 | WFS_PIN_BEEP_INACTIVE_AVAILABLE | Automatic beep tone on inactive key keypress is supported. If this flag is not set then automatic beeping for inactive keys is not supported. |
| 0x0008 | WFS_PIN_BEEP_INACTIVE_SELECTABLE | Automatic beeping for inactive keys can be controlled (i.e. turned on and off) by the application. If this flag is not set then automatic beeping for inactive keys cannot be controlled by an application. |

xfsPINCapabilitiesHSMVendor (17)

Specifies the name of the HSM vendor as a DisplayString.

xfsPINCapabilitiesHSMJournaling (18)

Specifies whether the HSM supports journaling by the WFS_CMD_PIN_GET_JOURNAL command as a TruthValue variable.

| Value | Meaning |
|---|---|
| True (1) | HSM supports journaling by WFS_CMD_PIN_GET_JOURNAL. |
| False (2) | HSM does not supports journaling by WFS_CMD_PIN_GET_JOURNAL. |

xfsPINCapabilitiesRSAAuthenticationScheme (19)

Specifies which type(s) of Remote Key Loading/Authentication is supported. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000001 | WFS_PIN_RSA_AUTH_2PARTY_SIG | Two-party Signature based authentication. |
| 0x00000002 | WFS_PIN_RSA_AUTH_3PARTY_CERT | Three-party Certificate based authentication. |

xfsPINCapabilitiesRSASignatureAlgorithm (20)

Specify which type(s) of RSA Signature Algorithm(s) is supported. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000001 | WFS_PIN_SIGN_RSASSA_PKCS1_V1_5 | SSA_PKCS_V1_5 Signatures supported. |
| 0x00000002 | WFS_PIN_SIGN_RSASSA_PSS | SSA_PSS Signatures supported. |

xfsPINCapabilitiesRSACryptAlgorithm (21)

Specify which type(s) of RSA Encipherment Algorithm(s) is supported. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000001 | WFS_PIN_CRYPT_RSAES_PKCS1_V1_5 | AES_PKCS_V1_5 algorithm supported. |
| 0x00000002 | WFS_PIN_CRYPT_RSAES_OAEP | AES_OAEP algorithm supported. |

xfsPINCapabilitiesRSAKeyCheckMode (22)

Specifies which algorithm/method used to generate the public key check value/thumb print. Values are a combination of hex values according to the values in the following table.

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000001 | WFS_PIN_RSA_KCV_SHA1 | SHA-1 is supported. |

xfsPINCapabilitiesSignatureScheme (23)

Specifies which capabilities are supported by the Signature scheme as one of the following flags as integer value. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x00000001 | WFS_PIN_SIG_GEN_RSA_KEY_PAIR | Specifies if the Service Provider supports the RSA Signature Scheme WFS_CMD_PIN_GENERATE_RSA_KEY_PAIR and WFS_CMD_PIN_EXPORT_RSA_EPP_SIGNED commands. |
| 0x00000002 | WFS_PIN_SIG_RANDOM_NUMBER | Specifies if the Service Provider returns a random number from the WFS_CMD_PIN_START_KEY_EXCHANGE command within the RSA Signature Scheme. |
| 0x00000004 | WFS_PIN_SIG_EXPORT_EPP_ID | Specifies if the Service Provider supports exporting the EPP Security Item within the RSA Signature Scheme. |
| 0x00000008 | WFS_PIN_SIG_ENHANCED_RKL | Specifies that the Service Provider supports the Enhanced Signature Remote Key Scheme. This scheme allows the customer to manage their own public keys independently of the Signature Issuer. When this mode is supported then the key loaded signed with the Signature Issuer key is the host root public key $PK_{ROOT}$, rather than $PK_{HOST}$. |

Import Schemes

The PIN device class defines a list of import schemes that may be supported by the device. These are defined as a zero-terminated array of WORDs so that it can be easily extended for future import schemes. The WORD values are sequential (1, 2, 3 …) and therefore cannot be combined (ORed) together to form a single unique value that indicates what import schemes the device supports. For this MIB definition entry, hex values have been defined for each XFS import scheme literal.

xfsPINCapabilitiesEMVImportSchemes (24)

Specifies Import Scheme(s) supported by the PIN Service. Values are a combination of hex values according to the values in the following table:

| Value | XFS Literal Name | XFS Literal Value |
|---|---|---|
| 0x0001 | WFS_PIN_EMV_IMPORT_PLAIN_CA | 0x0001 |

| | | |
|---|---|---|
| 0x0002 | WFS_PIN_EMV_IMPORT_CHKSUM_CA | 0x0002 |
| 0x0004 | WFS_PIN_EMV_IMPORT_EPI_CA | 0x0003 |
| 0x0008 | WFS_PIN_EMV_IMPORT_ISSUER | 0x0004 |
| 0x0010 | WFS_PIN_EMV_IMPORT_ICC | 0x0005 |
| 0x0020 | WFS_PIN_EMV_IMPORT_ICC_PIN | 0x0006 |
| 0x0040 | WFS_PIN_EMV_IMPORT_PKCSV1_5_CA | 0x0007 |

xfsPINCapabilitiesEMVHashAlgorithm (25)

Specifies which hash algorithm is supported for the calculation of the HASH; can be a combination of the following flags as a numeric value. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_HASH_SHA1_DIGEST | The SHA 1 digest algorithm is supported by the WFS_CMD_PIN_DIGEST command. |

xfsPINCapabilitiesKeyImportThroughParts (26)

Specifies whether the device is capable of importing keys in multiple parts as a TruthValue variable.

| Value | Meaning |
|---|---|
| True (1) | Device supports the key import in multiple parts. |
| False (2) | Device does not support key import in multiple parts. |

xfsPINCapabilitiesENCIOProtocols (27)

Specifies the ENC_IO protocols supported to communicate with the encryption module. Values are a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_ENC_PROT_CH | For Swiss specific protocols. The document specification for Swiss specific protocols is "CMD_ENC_IO - CH Protocol.doc". |
| 0x0002 | WFS_PIN_ENC_PROT_GIECB | Protocol for “Groupement des Cartes Bancaires” (France). |
| 0x0004 | WFS_PIN_ENC_PROT_LUX | Protocol for Luxemburg commands. The reference for this specific protocol is the Authorization Center in Luxemburg(CETREL.) Cryptography Management. |

xfsPINCapabilitiesTypeCombined (28)

Specifies whether the keypad used in the secure PIN pad module is integrated within a generic Win32 keyboard as a TruthValue.

| Value | Meaning |
|---|---|
| True(1) | The secure PIN keypad is integrated within a generic Win32 keyboard and standard Win32 key events will be generated for any key when there is no ‘active’ GET_DATA or GET_PIN command. Note that XFS continues to support defined PIN keys only, and is not extended to support new alphanumeric keys. |
| False(2) | The secure PIN keypad is not integrated within a generic Win32 keyboard. |

xfsPINCapabilitiesSetPinblockDataRequired (29)

Specifies whether the command WFS_CMD_PIN_SET_PINBLOCK_DATA must be called before the PIN is entered via WFS_CMD_PIN_GET_PIN and retrieved via WFS_CMD_PIN_GET_PINBLOCK as a TruthValue.

| Value | Meaning |
|---|---|

| | |
|---|---|
| True(1) | WFS_CMD_PIN_SET_PINBLOCK_DATA must be called before the PIN is entered via WFS_CMD_PIN_GET_PIN and retrieved via WFS_CMD_PIN_GET_PINBLOCK. |
| False(2) | WFS_CMD_PIN_SET_PINBLOCK_DATA is not called. |

xfsPINCapabilitiesKeyBlockImportFormats (30)

Supported key block formats; a combination of hex values according to the values in the following table:

| Value | XFS Name | Meaning |
|---|---|---|
| 0x0001 | WFS_PIN_ANSTR31KEYBLOCK | Supports ANS TR-31 Keyblock format key import. |

xfsPINCapabilitiesPowerSaveControl (31)

It contains the capability of the power saving control. It is a TruthValue type field. Allowed values are:

| Value | Meaning |
|---|---|
| True (1) | Power saving control is available. |
| False (2) | Power saving control is not available. |

xfsPINCapabilitiesExtraCapability (100)

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

# 3 PIN Traps

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

## 3.1 PIN 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 PIN 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 PIN reflect the PIN 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_PIN_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 104 defines the trap as a PIN Detailed Device Status Change trap. In the following section, the numbers in parenthesis at the end of each binding just indicate the sequence of the variable bindings within the trap, they do not represent an OID value.

### 3.1.1 PIN Detailed Device Status Change Trap Format

The following defines the variable bindings included in the PIN Detailed Device Status Change Trap.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName (1)
This variable binding contains the system generating the alarm, it is a Display String field. It corresponds to *lpszWorkstationName* in the device status change event data from the Service Provider.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName (2)
This variable binding represents the managed service name generating the alarm, it is a Display String field. The agent derives this field from the device status change event.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass (3)
This variable binding represents the XFS service class identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<Managed*ServiceName*>\ class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CDM=3, etc. See the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document for a complete list of these values.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName (4)
This variable binding represents the XFS service class name generating the alarm, it is a Display String field. It corresponds to the three character representation of the XFS device class name, and it is useful for human interpretation of a trap. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\ class.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType (5)
This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS_INF_PIN_CAPABILITIES.*fwType* field.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid (6)
This variable binding represents the OID of the sub-tree within *xfsManagedService* defining the management information for this class of managed service. This variable, along with the managed service name as an index, prevents the need for additional querying to find the service specific MIB branch. The PIN MIB class is represented by .1.3.6.1.4.1.16213.2.4

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.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevice**.xfsPINStatusManagedServiceName (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.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusNumberSubDevices.**xfsPINStatusManagedServiceName (14)
Defines how many sub-devices the service has.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusEncStat.**xfsPINStatusManagedServiceName (15)
It contains the encryptor module state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusExtraStatus.**xfsPINStatusManagedServiceName (16)
It contains the vendor dependent additional device status information as an OCTET STRING. The information is returned as a series of *"key=value"* strings. Each string is null-terminated, with the final string terminating with two null characters.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusGuidancePinPad.**xfsPINStatusManagedServiceName (17)
It contains the state of the guidance light indicator on the PIN pad unit. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusAutoBeepMode.**xfsPINStatusManagedServiceName (18)
It contains whether the PIN device will emit a key beep tone on key presses. It is a TruthValue type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusCertificateState.**xfsPINStatusManagedServiceName (19)

It contains the certificate state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevicePosition.**xfsPINStatusManagedServiceName (20)

It contains the device position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusPowerSaveRecoveryTime.**xfsPINStatusManagedServiceName (21)

It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. It is a numeric type field.

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

As an example, the following variable binding list represents a detailed device status change trap (6, 104) that is generated for a PIN with a managed service name of "PinPad1". It reports that the device is OFFLINE because the encryption module is not ready.

| | |
|---|---|
| xfsMIBRoot.3.1.3.1 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSysName) |
| | "SST System 1" |
| | |
| xfsMIBRoot.3.1.3.2 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName) |
| | "PinPad1" |
| | |
| xfsMIBRoot.3.1.3.3 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass) |
| | 4 (WFS_SERVICE_CLASS_PIN) |
| | |
| xfsMIBRoot.3.1.3.4 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName) |
| | "PIN" |
| | |
| xfsMIBRoot.3.1.3.5 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType) |
| | 0x0007 (WFS_PIN_TYPEHSM \| WFS_PIN_TYPEEDM \| WFS_PIN_TYPEEPP) |
| | |
| xfsMIBRoot.3.1.3.6 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid) |
| | ".1.3.6.1.4.1.16213.2.4" |
| | |
| xfsMIBRoot.3.1.3.7 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName) |
| | "ABC Corp Pin Pad" |
| | |
| xfsMIBRoot.3.1.3.8 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor) |
| | "Best Devices Incorporated" |
| | |
| xfsMIBRoot.3.1.3.9 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion) |
| | "1.10" |
| | |
| xfsMIBRoot.3.1.3.10 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapEvent) |
| | 4 (WFS_SYSE_DEVICE_STATUS) |
| | |
| xfsMIBRoot.3.1.3.11 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate) |
| | "20/03/2003 15:40:53 -300" |
| | |
| xfsMIBRoot.3.1.3.12 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion) |
| | "1.23" |
| | |
| xfsMIBRoot.2.4.1.2.1.**3**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevice**.xfsPINStatusManagedServiceName) |
| | 2 (WFS_STAT_DEVOFFLINE) |

| | |
|---|---|
| | |
| xfsMIBRoot.2.4.1.2.1.**2**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusNumberSubDevices**.xfsPINStatusManagedServiceName) |
| | 0 (No sub device) |
| | |
| xfsMIBRoot.2.4.1.2.1.**4**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusEncStat**.xfsPINStatusManagedServiceName) |
| | 2 (xfsPINEncNotReady) |
| | |
| xfsMIBRoot.2.4.1.2.1.**100**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusExtraStatus**.xfsPINStatusManagedServiceName) |
| | '\0"\0' ( No extra data ) |
| | |
| xfsMIBRoot.2.4.1.2.1.**5**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusGuidancePinPad**.xfsPINStatusManagedServiceName) |
| | 0 (value corresponding to WFS_PIN_GUIDANCE_NOT_AVAILABLE) |
| | |
| xfsMIBRoot.2.4.1.2.1.**6**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusAutoBeepMode**.xfsPINStatusManagedServiceName) |
| | 1 (value corresponding to WFS_PIN_BEEP_ON_ACTIVE) |
| | |
| xfsMIBRoot.2.4.1.2.1.**7**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusCertificateState**.xfsPINStatusManagedServiceName) |
| | 1 (xfsPINCertUnknown) |
| | |
| xfsMIBRoot.2.4.1.2.1.**8**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevicePosition**.xfsPINStatusManagedServiceName) |
| | 1 (xfsPINDeviceInPosition) |
| | |
| xfsMIBRoot.2.4.1.2.1.**9**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusPowerSaveRecoveryTime**.xfsPINStatusManagedServiceName) |
| | 3 (3 seconds to recover from power saving mode) |

## 3.2 PIN Sub-Device Status Change Trap

The PIN does not currently support any sub-devices so the PIN Sub-Device Status Change Trap is not currently defined. The SNMP Specific trap value 204 is reserved in case a sub-device is ever added to the PIN device class.

## 3.3 PIN Reset Device Complete Trap

On the PIN device class this trap reports the completion of the reset device request and includes the status of the device at that point. If the reset has changed the status of the device then the Device Status Change and a Detail Device Status traps will also be generated.

The SNMP Specific trap value 304 defines the trap as a PIN Reset Device Complete trap.

### 3.3.1 PIN Reset Device Complete Trap Format

The following defines the variable bindings included in the PIN Reset Device Complete Trap. In the following section, the numbers in parenthesis at the end of each binding just indicate the sequence of the variable bindings within the trap, they do not represent an OID value.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult (1)
This variable binding contains a value indicating if the reset was executed, and if not provides a reason. It does not report the status of the device (i.e. the result of the reset), the current status of the device is reported within the **xfsPINStatusDevice** binding (var bind 12 below).

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName (2)
This variable binding represents the managed service name generating the alarm, it is a Display String field. The agent derives this field from the device status change event.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass (3)

This variable binding represents the XFS service class identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the class identifier for the class name. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<Managed*ServiceName*>\ class. This ID matches the class OID branch number i.e. PTR=1, IDC=2, CDM=3, etc. See the *XFS MIB Architecture and SNMP Extensions Programmer's Reference* document for a complete list of these values.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName (4)
This variable binding represents the XFS service class name generating the alarm, it is a Display String field. It corresponds to the three character representation of the XFS device class name, and it is useful for human interpretation of a trap. The class name is identified from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\ class.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType (5)
This variable binding represents the XFS type identifier generating the alarm, it is a 32-bit integer (INT32). It corresponds to the type identifier as defined in the WFS_INF_PIN_CAPABILITIES.*fwType* field.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid (6)
This variable binding represents the OID of the sub-tree within *xfsManagedService* defining the management information for this class of managed service. This variable, along with the managed service name as an index, prevents the need for additional querying to find the service specific MIB branch. The PIN MIB class is represented by .1.3.6.1.4.1.16213.2.4

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceName (7)
This variable binding represents the physical device name or names associated with the managed service generating the alarm, it is a Display String field. It corresponds to the physical device name or names identified by the managed service. The managed service name is used to identify the physical device name or names, from registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<ManagedServiceName>\ PhysicalDeviceName. Multiple physical device names are comma separated.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor (8)
This variable binding represents the XFS device vendor name of the device generating the alarm, it is a Display String field. It corresponds to the vendor name for the Service Provider. The Service Provider is identified from the managed service name and the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\ ServiceProvider.
The Service Provider name is then used to identify the vendor, from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS\<*ServiceProviderName*>\vendor_ name.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion (9)
This variable binding represents the XFS MIB version of the device generating the alarm, it is a Display String field. It corresponds to the XFS MIB version for the managed service. The managed service name is used to identify the XFS MIB version, from registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\ MibVersion.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate (10)
This variable represents the UTC and bias for local translation of the date and time when the event was generated. It is a Display String field. The data is formatted in the following way: "DD/MM/YYYY HH:MM:SS +ZZZ" where DD/MM/YYYY HH:MM:SS is the local date and time. ZZZ is the bias, which is the difference, in minutes, between Co-ordinated Universal Time (UTC) and local time.

xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion (11)
This variable represents the vendor-defined version of the Service Provider generating the alarm, it is a Display String field. The Service Provider is identified from the managed service name and the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\MANAGEMENT_PROVIDERS\<*ManagedServiceName*>\ ServiceProvider.
The Service Provider name is then used to identify the version, from the registry value HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS\<*ServiceProviderName*>\version.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevice**.
xfsPINStatusManagedServiceName (12)
This variable binding represents the current state of the physical device managed by the service. It is a 32 bit integer (INT32).

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusNumberSubDevices.**xfsPINStatusManagedServiceName (13)
Defines how many sub-devices the service has. This is the number of retract bins the device supports.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusEncStat.**xfsPINStatusManagedServiceName (14)
It contains the encryptor module state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusExtraStatus.**xfsPINStatusManagedServiceName (15)
It contains the vendor dependent additional device status information as an OCTET STRING. The information is returned as a series of *"key=value"* strings. Each string is null-terminated, with the final string terminating with two null characters.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusGuidancePinPad.**xfsPINStatusManagedServiceName (16)
It contains the state of the guidance light indicator on the PIN pad unit. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusAutoBeepMode.**xfsPINStatusManagedServiceName (17)
It contains whether the PIN device will emit a key beep tone on key presses. It is a TruthValue type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusCertificateState.**xfsPINStatusManagedServiceName (18)
It contains the certificate state. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevicePosition.**xfsPINStatusManagedServiceName (19)
It contains the device position. It is a numeric type field.

xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusPowerSaveRecoveryTime.**xfsPINStatusManagedServiceName (20)
It contains the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. It is a numeric type field.

### 3.3.2 PIN Reset Device Complete: an example

As an example, the following variable binding list represents a Reset Device Complete trap (6, 304) generated as the result of a request to reset the device from the remote management station. The device in question has a managed service name "PinPad1".

| | |
|---|---|
| xfsMIBRoot.3.1.3.13 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapResetDeviceResult) |
| | 0 (resetExecuted) |
| | |
| xfsMIBRoot.3.1.3.2 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceName) |
| | "PinPad1" |
| | |
| xfsMIBRoot.3.1.3.3 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClass) |
| | 4 (WFS_SERVICE_CLASS_PIN) |
| | |
| xfsMIBRoot.3.1.3.4 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceClassName) |
| | "PIN" |
| | |
| xfsMIBRoot.3.1.3.5 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceType) |
| | 0x0007 (WFS_PIN_TYPEHSM \| WFS_PIN_TYPEEDM \| WFS_PIN_TYPEEPP) |
| | |
| xfsMIBRoot.3.1.3.6 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapManagedServiceOid) |
| | ".1.3.6.1.4.1.16213.2.4" |
| | |
| xfsMIBRoot.3.1.3.7 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapPhysicalDeviceNam |

| | |
|---|---|
| | e) |
| | “ABC Corp Pin Pad” |
| | |
| xfsMIBRoot.3.1.3.8 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDeviceVendor) |
| | “Best Devices Incorporated” |
| | |
| xfsMIBRoot.3.1.3.9 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapMIBVersion) |
| | “1.10” |
| | |
| xfsMIBRoot.3.1.3.11 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapDate) |
| | “20/03/2003 15:40:53 -300” |
| | |
| xfsMIBRoot.3.1.3.12 | (xfsMIBRoot.xfsTrap.xfsTrapV1.xfsCommonTrapVars.xfsCommonTrapSPVersion) |
| | “1.23” |
| | |
| xfsMIBRoot.2.4.1.2.1.**3**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevice**.xfsPINStatusManagedServiceName) |
| | 2 (WFS_STAT_DEVOFFLINE) |
| | |
| xfsMIBRoot.2.4.1.2.1.**2**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusNumberSubDevices**.xfsPINStatusManagedServiceName) |
| | 0 (No sub device) |
| | |
| xfsMIBRoot.2.4.1.2.1.**4**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusEncStat**.xfsPINStatusManagedServiceName) |
| | 2 (xfsPINEncNotReady) |
| | |
| xfsMIBRoot.2.4.1.2.1.**100**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusExtraStatus**.xfsPINStatusManagedServiceName) |
| | '\0"\0' ( No extra data ) |
| | |
| xfsMIBRoot.2.4.1.2.1.**5**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusGuidancePinPad**.xfsPINStatusManagedServiceName) |
| | 0 (value corresponding to WFS_PIN_GUIDANCE_NOT_AVAILABLE) |
| | |
| xfsMIBRoot.2.4.1.2.1.**6**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusAutoBeepMode**.xfsPINStatusManagedServiceName) |
| | 1 (TRUE) |
| | |
| xfsMIBRoot.2.4.1.2.1.**7**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusCertificateState**.xfsPINStatusManagedServiceName) |
| | 1 (xfsPINCertUnknown) |
| | |
| xfsMIBRoot.2.4.1.2.1.**8**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusDevicePosition**.xfsPINStatusManagedServiceName) |
| | 1 (xfsPINDeviceInPosition) |
| xfsMIBRoot.2.4.1.2.1.**9**.Index | (xfsMIBRoot.xfsManagedService.xfsPIN.xfsPINV1.xfsPINStatusTable.xfsPINStatusEntry.**xfsPINStatusPowerSaveRecoveryTime**.xfsPINStatusManagedServiceName) |
| | 3 (3 seconds to recover from power saving mode) |

# 4 Appendix A - PIN MIB sub-tree

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

## 4.1 PIN MIB in SMIv2 and SMIv1 ASN-1 format


SMIv1_xfsPIN.mib


SMIv2_xfsPIN.mib

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

```
-- ****************************************************************
-- XFS MIB for PIN
-- Management Information Base for XFS PIN Device
--
-- The PIN Number is 4
-- The ASN.1 prefix to, and including the PIN is: 1.3.6.1.4.1.16213.2.4
--
-- ****************************************************************

  XFS-PIN-MIB DEFINITIONS ::= BEGIN

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


--
-- Type definitions
--

--  ****************************************************************
-- PIN #defines
-- ****************************************************************
    IxfsPINEncStatStatus ::= INTEGER
      {
      xfsPINEncReady(1),
      xfsPINEncNotReady(2),
      xfsPINEncNotInitialized(3),
      xfsPINEncBusy(4),
      xfsPINEncundefined(5),
      xfsPINEncInitialized(6)
      }

    IxfsPINCertificateStateStatus ::= INTEGER
      {
      xfsPINCertUnknown(1),
      xfsPINCertPrimary(2),
      xfsPINCertSecondary(3),
      xfsPINCertNotReady(5)
      }

    IxfsPINDevicePositionStatus ::= INTEGER
      {
      xfsPINDeviceInPosition(1),
      xfsPINDeviceNotInPosition(2),
      xfsPINDevicePosUnknown(3),
      xfsPINDevicePosNotSupported(4)
      }

    IxfsPINCapabilitiesDisplayType ::= INTEGER
      {
      xfsPINDispNone(2),
```

```
      xfsPINDispLedThrough(3),
      xfsPINDispDisplay(4)
      }



--
-- Node definitions
--

--  ********************************************************************
-- Version 1 of PIN MIB
--
-- The ASN.1 prefix to, and including the Version 1 of PIN is:
               1.3.6.1.4.1.16213.2.4.1
--
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1
    xfsPINV1 OBJECT IDENTIFIER ::= { xfsPIN 1 }


    -- 1.3.6.1.4.1.16213.2.4.1.1
    xfsPINInstances OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Number that represents the number of PIN managed services."
      ::= { xfsPINV1 1 }


--  ************************************************************************
-- PIN Device Status Table
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1.2
    xfsPINStatusTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsPINStatusEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the PIN status table."
      ::= { xfsPINV1 2 }


    -- 1.3.6.1.4.1.16213.2.4.1.2.1
    xfsPINStatusEntry OBJECT-TYPE
      SYNTAX XfsPINStatusEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "PIN Device Status Table Entry."
      INDEX { xfsPINStatusManagedServiceName }
      ::= { xfsPINStatusTable 1 }


    XfsPINStatusEntry ::=
      SEQUENCE {
        xfsPINStatusManagedServiceName
          DisplayString,
        xfsPINStatusNumberSubDevices
          Integer32,
        xfsPINStatusDevice
          IxfsMIBDeviceStatus,
        xfsPINStatusEncStat
          IxfsPINEncStatStatus,
        xfsPINStatusGuidancePinPad
          Integer32,
        xfsPINStatusAutoBeepMode
          Integer32,
        xfsPINStatusCertificateState
          IxfsPINCertificateStateStatus,
```

```
    xfsPINStatusDevicePosition
      IxfsPINDevicePositionStatus,
    xfsPINStatusPowerSaveRecoveryTime
      Integer32,
    xfsPINStatusExtraStatus
      OCTET STRING
   }

-- 1.3.6.1.4.1.16213.2.4.1.2.1.1
xfsPINStatusManagedServiceName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Instance identifier of the managed service."
  ::= { xfsPINStatusEntry 1 }


-- 1.3.6.1.4.1.16213.2.4.1.2.1.2
xfsPINStatusNumberSubDevices OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Number of sub devices supported by the PIN device."
  ::= { xfsPINStatusEntry 2 }


-- 1.3.6.1.4.1.16213.2.4.1.2.1.3
xfsPINStatusDevice OBJECT-TYPE
  SYNTAX IxfsMIBDeviceStatus
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Device status."
  ::= { xfsPINStatusEntry 3 }


-- 1.3.6.1.4.1.16213.2.4.1.2.1.4
xfsPINStatusEncStat OBJECT-TYPE
  SYNTAX IxfsPINEncStatStatus
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Encryption Module status.
    xfsPINEncReady (1),
    xfsPINEncNotReady (2),
    xfsPINEncNotInitialized (3),
    xfsPINEncBusy (4),
    xfsPINEncundefined (5),
    xfsPINEncInitialized (6)"
  ::= { xfsPINStatusEntry 4 }


-- 1.3.6.1.4.1.16213.2.4.1.2.1.5
xfsPINStatusGuidancePinPad OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "state of the guidance light indicator."
  ::= { xfsPINStatusEntry 5 }


-- 1.3.6.1.4.1.16213.2.4.1.2.1.6
xfsPINStatusAutoBeepMode OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Automatic beep tone on key press is active or not.
```

```
        xfsPINStatusBeepOnActive(1),
        XfsPINStatusBeepOnInActive(2)"
      ::= { xfsPINStatusEntry 6 }


    -- 1.3.6.1.4.1.16213.2.4.1.2.1.7
    xfsPINStatusCertificateState OBJECT-TYPE
      SYNTAX IxfsPINCertificateStateStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Certificate status. xfsPINCertNotSupp   (1), xfsPINCertPrimary (2),
                 xfsPINCertSecondary (3), xfsPINCertNotReady (5)"
      ::= { xfsPINStatusEntry 7 }


    -- 1.3.6.1.4.1.16213.2.4.1.2.1.8
    xfsPINStatusDevicePosition OBJECT-TYPE
      SYNTAX IxfsPINDevicePositionStatus
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the device position.
        xfsPINDeviceInPosition(1),
        xfsPINDeviceNotInPosition(2),
        xfsPINDevicePosUnknown(3),
        xfsPINDevicePosNotSupported(4)."
      ::= { xfsPINStatusEntry 8 }


    -- 1.3.6.1.4.1.16213.2.4.1.2.1.9
    xfsPINStatusPowerSaveRecoveryTime OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Specifies the actual number of seconds required by the device to resume
                 its normal operational state from the current power saving mode.
                 This value is zero if either the power saving mode has not been
                 activated or no power save control is supported."
      ::= { xfsPINStatusEntry 9 }


    -- 1.3.6.1.4.1.16213.2.4.1.2.1.100
    xfsPINStatusExtraStatus OBJECT-TYPE
      SYNTAX OCTET STRING
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Vendor dependent additional device status information."
      ::= { xfsPINStatusEntry 100 }


--  ********************************************************************
-- PIN Sub Device Status Table
--
-- Note that the PIN 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 16213.2.4.1.3
-- must be reserved for the sub-device table.
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1.3
    xfsPINSubDeviceTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsPINSubDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the PIN Sub-Device Status Table."
      ::= { xfsPINV1 3 }
```

```
    -- 1.3.6.1.4.1.16213.2.4.1.3.1
    xfsPINSubDeviceEntry OBJECT-TYPE
      SYNTAX XfsPINSubDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "PIN Sub-Device Status Table Entry."
      INDEX { xfsPINSubDeviceManagedServiceName, xfsPINSubDeviceIndex }
      ::= { xfsPINSubDeviceTable 1 }


    XfsPINSubDeviceEntry ::=
      SEQUENCE {
        xfsPINSubDeviceManagedServiceName
          DisplayString,
        xfsPINSubDeviceIndex
          INTEGER
       }

--    As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
--   xfsPINSubDeviceValue             INTEGER }
    -- 1.3.6.1.4.1.16213.2.4.1.3.1.1
    xfsPINSubDeviceManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsPINSubDeviceEntry 1 }


    -- 1.3.6.1.4.1.16213.2.4.1.3.1.2
    xfsPINSubDeviceIndex OBJECT-TYPE
      SYNTAX INTEGER (1..65535)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Index into the array of sub devices supported."
      ::= { xfsPINSubDeviceEntry 2 }


--    As an example if you want to add values to the sub-device table, add
-- entries as shown in the example below.
-- xfsPINSubDeviceValue  OBJECT-TYPE
--   SYNTAX        INTEGER
--   ACCESS        read-only
--   STATUS        mandatory
--   DESCRIPTION   "Returns the value of the sub device referenced by the index."
--   ::= {xfsPINSubDeviceEntry 3}
-- ****************************************************************
-- PIN Error Table
-- ****************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1.4
    xfsPINErrorTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsPINErrorEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the PIN Error Table."
      ::= { xfsPINV1 4 }


    -- 1.3.6.1.4.1.16213.2.4.1.4.1
    xfsPINErrorEntry OBJECT-TYPE
      SYNTAX XfsPINErrorEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
```

```
        "PIN Error Table Entry."
      INDEX { xfsPINErrorManagedServiceName, xfsPINErrorCommandCode,
                xfsPINErrorResponseCode }
      ::= { xfsPINErrorTable 1 }


    XfsPINErrorEntry ::=
      SEQUENCE {
        xfsPINErrorManagedServiceName
          DisplayString,
        xfsPINErrorCommandCode
          INTEGER,
        xfsPINErrorResponseCode
          INTEGER,
        xfsPINErrorCount
          Integer32
       }

    -- 1.3.6.1.4.1.16213.2.4.1.4.1.1
    xfsPINErrorManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsPINErrorEntry 1 }


    -- 1.3.6.1.4.1.16213.2.4.1.4.1.2
    xfsPINErrorCommandCode OBJECT-TYPE
      SYNTAX INTEGER (401..500)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "The executable command code supported by the Service
        Provider associated with the error count of interest."
      ::= { xfsPINErrorEntry 2 }


    -- 1.3.6.1.4.1.16213.2.4.1.4.1.3
    xfsPINErrorResponseCode OBJECT-TYPE
      SYNTAX INTEGER (0..99 | 400..499)
      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."
      ::= { xfsPINErrorEntry 3 }


    -- 1.3.6.1.4.1.16213.2.4.1.4.1.4
    xfsPINErrorCount OBJECT-TYPE
      SYNTAX Integer32
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "The counter value corresponding to the managed service,
        command code and response code."
      ::= { xfsPINErrorEntry 4 }


--  ********************************************************************
-- PIN Reset Table
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1.5
    xfsPINResetTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsPINResetEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
```

```
        "Defines the set of MIB Variables for the PIN Reset Table."
      ::= { xfsPINV1 5 }


    -- 1.3.6.1.4.1.16213.2.4.1.5.1
    xfsPINResetEntry OBJECT-TYPE
      SYNTAX XfsPINResetEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "PIN Reset Table Entry."
      INDEX { xfsPINResetManagedServiceName }
      ::= { xfsPINResetTable 1 }


    XfsPINResetEntry ::=
      SEQUENCE {
        xfsPINResetManagedServiceName
          DisplayString,
        xfsPINResetAll
          Integer32,
        xfsPINResetTimestamp
          DisplayString
       }

    -- 1.3.6.1.4.1.16213.2.4.1.5.1.1
    xfsPINResetManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsPINResetEntry 1 }


    -- 1.3.6.1.4.1.16213.2.4.1.5.1.2
    xfsPINResetAll 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."
      ::= { xfsPINResetEntry 2 }


    -- 1.3.6.1.4.1.16213.2.4.1.5.1.3
    xfsPINResetTimestamp OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Date and time the last reset of the counters was
        performed."
      ::= { xfsPINResetEntry 3 }


--   ********************************************************************
-- PIN Reset Device Table
-- ********************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1.6
    xfsPINResetDeviceTable OBJECT-TYPE
      SYNTAX SEQUENCE OF XfsPINResetDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "Define the set of MIB Variables for the PIN Reset Device Table."
      ::= { xfsPINV1 6 }


    -- 1.3.6.1.4.1.16213.2.4.1.6.1
```

```
    xfsPINResetDeviceEntry OBJECT-TYPE
      SYNTAX XfsPINResetDeviceEntry
      MAX-ACCESS not-accessible
      STATUS current
      DESCRIPTION
        "PIN Reset Device Table Entry."
      INDEX { xfsPINResetDeviceManagedServiceName }
      ::= { xfsPINResetDeviceTable 1 }


    XfsPINResetDeviceEntry ::=
      SEQUENCE {
        xfsPINResetDeviceManagedServiceName
          DisplayString,
        xfsPINResetDeviceAction
          INTEGER,
        xfsPINResetDeviceMediaControl
          INTEGER,
        xfsPINResetDeviceStatus
          INTEGER
       }

    -- 1.3.6.1.4.1.16213.2.4.1.6.1.1
    xfsPINResetDeviceManagedServiceName OBJECT-TYPE
      SYNTAX DisplayString
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Instance identifier of the managed service."
      ::= { xfsPINResetDeviceEntry 1 }


    -- 1.3.6.1.4.1.16213.2.4.1.6.1.2
    xfsPINResetDeviceAction OBJECT-TYPE
      SYNTAX INTEGER { executeReset(1) }
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
        "Variable that initiates the device reset."
      ::= { xfsPINResetDeviceEntry 2 }


    -- 1.3.6.1.4.1.16213.2.4.1.6.1.3
    xfsPINResetDeviceMediaControl OBJECT-TYPE
      SYNTAX INTEGER { mediaDefault(1) }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Variable that reports the media handling during the device reset."
      ::= { xfsPINResetDeviceEntry 3 }


    -- 1.3.6.1.4.1.16213.2.4.1.6.1.4
    xfsPINResetDeviceStatus OBJECT-TYPE
      SYNTAX INTEGER
        {
        resetIdle(1),
        resetInProgress(2)
        }
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
        "Variable that reports the  progress of the device reset."
      ::= { xfsPINResetDeviceEntry 4 }


--    ************************************************************************
-- PIN Device Capabilities Table
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.2.4.1.7
    xfsPINCapabilitiesTable OBJECT-TYPE
```

```
  SYNTAX SEQUENCE OF XfsPINCapabilitiesEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Define the set of MIB Variables for the PIN Capabilities table."
  ::= { xfsPINV1 7 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1
xfsPINCapabilitiesEntry OBJECT-TYPE
  SYNTAX XfsPINCapabilitiesEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "PIN Device Capabilities Table Entry."
  INDEX { xfsPINCapabilitiesManagedServiceName }
  ::= { xfsPINCapabilitiesTable 1 }


XfsPINCapabilitiesEntry ::=
  SEQUENCE {
    xfsPINCapabilitiesManagedServiceName
      DisplayString,
    xfsPINCapabilitiesDeviceType
      Integer32,
    xfsPINCapabilitiesCompoundDevice
      TruthValue,
    xfsPINCapabilitiesKeyNumber
      Integer32,
    xfsPINCapabilitiesAlgorithm
      Integer32,
    xfsPINCapabilitiesPinFormats
      Integer32,
    xfsPINCapabilitiesDerivationAlgorithms
      Integer32,
    xfsPINCapabilitiesPresentationAlgorithms
      Integer32,
    xfsPINCapabilitiesDisplay
      IxfsPINCapabilitiesDisplayType,
    xfsPINCapabilitiesConnect
      TruthValue,
    xfsPINCapabilitiesIDKey
      Integer32,
    xfsPINCapabilitiesValidationAlgorithms
      Integer32,
    xfsPINCapabilitiesKeyCheckModes
      Integer32,
    xfsPINCapabilitiesGuidancePinPad
      Integer32,
    xfsPINCapabilitiesPINCanPersistAfterUse
      TruthValue,
    xfsPINCapabilitiesAutoBeep
      Integer32,
    xfsPINCapabilitiesHSMVendor
      DisplayString,
    xfsPINCapabilitiesHSMJournaling
      TruthValue,
    xfsPINCapabilitiesRSAAuthenticationScheme
      Integer32,
    xfsPINCapabilitiesRSASignatureAlgorithm
      Integer32,
    xfsPINCapabilitiesRSACryptAlgorithm
      Integer32,
    xfsPINCapabilitiesRSAKeyCheckMode
      Integer32,
    xfsPINCapabilitiesSignatureScheme
      Integer32,
    xfsPINCapabilitiesEMVImportSchemes
      Integer32,
    xfsPINCapabilitiesEMVHashAlgorithm
      Integer32,
```

```
    xfsPINCapabilitiesKeyImportThroughParts
      TruthValue,
    xfsPINCapabilitiesENCIOProtocols
      Integer32,
    xfsPINCapabilitiesTypeCombined
      TruthValue,
    xfsPINCapabilitiesSetPinblockDataRequired
      TruthValue,
    xfsPINCapabilitiesKeyBlockImportFormats
      Integer32,
    xfsPINCapabilitiesPowerSaveControl
      TruthValue,
    xfsPINCapabilitiesExtraCapability
      OCTET STRING
   }

-- 1.3.6.1.4.1.16213.2.4.1.7.1.1
xfsPINCapabilitiesManagedServiceName OBJECT-TYPE
  SYNTAX DisplayString
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Instance identifier of the managed service."
  ::= { xfsPINCapabilitiesEntry 1 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.2
xfsPINCapabilitiesDeviceType OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "The type of the PIN device."
  ::= { xfsPINCapabilitiesEntry 2 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.3
xfsPINCapabilitiesCompoundDevice OBJECT-TYPE
  SYNTAX TruthValue
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies whether the logical device is part of a compound physical
             device."
  ::= { xfsPINCapabilitiesEntry 3 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.4
xfsPINCapabilitiesKeyNumber OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies the number of the keys which can be stored in the
             encryption/decryption module."
  ::= { xfsPINCapabilitiesEntry 4 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.5
xfsPINCapabilitiesAlgorithm OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Supported encryption modes."
  ::= { xfsPINCapabilitiesEntry 5 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.6
xfsPINCapabilitiesPinFormats OBJECT-TYPE
  SYNTAX Integer32
```

```
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Supported pin formats."
  ::= { xfsPINCapabilitiesEntry 6 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.7
xfsPINCapabilitiesDerivationAlgorithms OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Supported derivation algorithms."
  ::= { xfsPINCapabilitiesEntry 7 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.8
xfsPINCapabilitiesPresentationAlgorithms OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Supported presentation algorithms."
  ::= { xfsPINCapabilitiesEntry 8 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.9
xfsPINCapabilitiesDisplay OBJECT-TYPE
  SYNTAX IxfsPINCapabilitiesDisplayType
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies the type of the display used in the PIN pad module."
  ::= { xfsPINCapabilitiesEntry 9 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.10
xfsPINCapabilitiesConnect OBJECT-TYPE
  SYNTAX TruthValue
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies whether the PIN pad is directly physically connected to the ID
             card Unit."
  ::= { xfsPINCapabilitiesEntry 10 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.11
xfsPINCapabilitiesIDKey OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies whether an ID key is supported."
  ::= { xfsPINCapabilitiesEntry 11 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.12
xfsPINCapabilitiesValidationAlgorithms OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies the algorithms for PIN validation supported by the service."
  ::= { xfsPINCapabilitiesEntry 12 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.13
xfsPINCapabilitiesKeyCheckModes OBJECT-TYPE
  SYNTAX Integer32
```

```
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies the key check modes that are supported to check the correctness
               of an imported key value."
    ::= { xfsPINCapabilitiesEntry 13 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.14
  xfsPINCapabilitiesGuidancePinPad OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "It contains the capability of the pinpad guidance light."
    ::= { xfsPINCapabilitiesEntry 14 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.15
  xfsPINCapabilitiesPINCanPersistAfterUse OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether the device can retain the PIN after a pin processing
               command."
    ::= { xfsPINCapabilitiesEntry 15 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.16
  xfsPINCapabilitiesAutoBeep OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether the PIN device will emit a key beep tone on key
               presses."
    ::= { xfsPINCapabilitiesEntry 16 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.17
  xfsPINCapabilitiesHSMVendor OBJECT-TYPE
    SYNTAX DisplayString
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "It identifies the HSM Vendor."
    ::= { xfsPINCapabilitiesEntry 17 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.18
  xfsPINCapabilitiesHSMJournaling OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether the HSM supports journaling by the
               WFS_CMD_PIN_GET_JOURNAL."
    ::= { xfsPINCapabilitiesEntry 18 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.19
  xfsPINCapabilitiesRSAAuthenticationScheme OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies which type(s) of Remote Key Loading/Authentication is
               supported."
    ::= { xfsPINCapabilitiesEntry 19 }
```

```
-- 1.3.6.1.4.1.16213.2.4.1.7.1.20
xfsPINCapabilitiesRSASignatureAlgorithm OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specify which type(s) of RSA Signature Algorithm(s) is supported."
  ::= { xfsPINCapabilitiesEntry 20 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.21
xfsPINCapabilitiesRSACryptAlgorithm OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specify which type(s) of RSA Encipherment Algorithm(s) is supported ."
  ::= { xfsPINCapabilitiesEntry 21 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.22
xfsPINCapabilitiesRSAKeyCheckMode OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies which algorithm/method used to generate the public key check
             value/thumb print."
  ::= { xfsPINCapabilitiesEntry 22 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.23
xfsPINCapabilitiesSignatureScheme OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies which capabilities are supported by the Signature scheme."
  ::= { xfsPINCapabilitiesEntry 23 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.24
xfsPINCapabilitiesEMVImportSchemes OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Identifies the supported EMV Import Scheme(s)."
  ::= { xfsPINCapabilitiesEntry 24 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.25
xfsPINCapabilitiesEMVHashAlgorithm OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies which hash algorithm is supported for the calculation of the
             HASH."
  ::= { xfsPINCapabilitiesEntry 25 }


-- 1.3.6.1.4.1.16213.2.4.1.7.1.26
xfsPINCapabilitiesKeyImportThroughParts OBJECT-TYPE
  SYNTAX TruthValue
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Specifies whether the device is capable of importing keys in multiple
             parts."
```

```
    ::= { xfsPINCapabilitiesEntry 26 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.27
  xfsPINCapabilitiesENCIOProtocols OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies the ENC_IO protocols supported to communicate with the
               encryption module."
    ::= { xfsPINCapabilitiesEntry 27 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.28
  xfsPINCapabilitiesTypeCombined OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether the keypad used in the secure PIN pad module is
               integrated within a generic Win32 keyboard."
    ::= { xfsPINCapabilitiesEntry 28 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.29
  xfsPINCapabilitiesSetPinblockDataRequired OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether the command WFS_CMD_PIN_SET_PINBLOCK_DATA must be called
               before the PIN is entered via WFS_CMD_PIN_GET_PIN and retrieved
               via WFS_CMD_PIN_GET_PINBLOCK."
    ::= { xfsPINCapabilitiesEntry 29 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.30
  xfsPINCapabilitiesKeyBlockImportFormats OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Supported key block formats."
    ::= { xfsPINCapabilitiesEntry 30 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.31
  xfsPINCapabilitiesPowerSaveControl OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Specifies whether power saving control is available."
    ::= { xfsPINCapabilitiesEntry 31 }


  -- 1.3.6.1.4.1.16213.2.4.1.7.1.100
  xfsPINCapabilitiesExtraCapability OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
      "Vendor dependent additional device Capabilities information."
    ::= { xfsPINCapabilitiesEntry 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
-- ************************************************************************
    -- 1.3.6.1.4.1.16213.3.0.104
    xfsPINDetailedDSCTrap NOTIFICATION-TYPE
      OBJECTS { xfsCommonTrapSysName, xfsCommonTrapManagedServiceName,
                xfsCommonTrapManagedServiceClass,
                xfsCommonTrapManagedServiceClassName,
                xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
                xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion,
                xfsCommonTrapEvent,
        xfsCommonTrapDate, xfsCommonTrapSPVersion, xfsPINStatusDevice,
                xfsPINStatusNumberSubDevices, xfsPINStatusEncStat,
        xfsPINStatusExtraStatus, xfsPINStatusGuidancePinPad,
                xfsPINStatusAutoBeepMode, xfsPINStatusCertificateState,
                xfsPINStatusDevicePosition,
        xfsPINStatusPowerSaveRecoveryTime }
      STATUS current
      DESCRIPTION
        "This trap indicates a change in the status of a managed
        service."
      ::= { xfsTrapV2 104 }


    -- 1.3.6.1.4.1.16213.3.0.304
    xfsPINResetDeviceCompleteTrap NOTIFICATION-TYPE
      OBJECTS { xfsCommonTrapResetDeviceResult, xfsCommonTrapManagedServiceName,
                xfsCommonTrapManagedServiceClass,
                xfsCommonTrapManagedServiceClassName,
                xfsCommonTrapManagedServiceType,
        xfsCommonTrapManagedServiceOid, xfsCommonTrapPhysicalDeviceName,
                xfsCommonTrapDeviceVendor, xfsCommonTrapMIBVersion,
                xfsCommonTrapDate,
        xfsCommonTrapSPVersion, xfsPINStatusDevice, xfsPINStatusNumberSubDevices,
                xfsPINStatusEncStat, xfsPINStatusExtraStatus,
        xfsPINStatusGuidancePinPad, xfsPINStatusAutoBeepMode,
                xfsPINStatusCertificateState, xfsPINStatusDevicePosition,
                xfsPINStatusPowerSaveRecoveryTime
         }
      STATUS current
      DESCRIPTION
        "This trap indicates the Reset action has complete and reports the
        state of the device after the reset."
      ::= { xfsTrapV2 304 }



  END

--
-- SMIv2_XFSPIN.mib
--
```

# 5 Appendix B - C-Header files

## 5.1 XFSMIBPIN.H

```
/******************************************************************************
*                                                                             *
* xfsmibpin.h      CEN/XFS - MIB PIN                                          *
*                                                                             *
*               Version 3.10  --  Dec 14, 2010                                *
*                                                                             *
******************************************************************************/

#ifndef __inc_xfsmibpin__h
#define __inc_xfsmibpin__h

#ifdef __cplusplus
extern "C" {
#endif

enum IxfsPINEncStatStatus
{
        xfsPINEncReady              = 1,
        xfsPINDevNotReady,
        xfsPINEncNotInitialised,
        xfsPINEncBusy,
        xfsPINEncUndefined,
        xfsPINEncInitialized

} xfsPINEncStatStatus;

enum IxfsPINCertificateStateStatus
{
        xfsPINCertUnknown           = 1,
        xfsPINCertPrimary           = 2,
        xfsPINCertSecondary         = 3,
        xfsPINCertNotReady          = 5

} xfsPINCertificateStateStatus;

enum IxfsPINDevicePositionStatus
{
        xfsPINDeviceInPosition      = 1,
        xfsPINDeviceNotInPosition,
        xfsPINDevicePosUnknown,
        xfsPINDevicePosNotSupported

} xfsPINDevicePositionStatus;

enum IxfsPINCapabilitiesDisplayType
{
        xfsPINDispNone              = 2,
        xfsPINDispLedThrough,
        xfsPINDispDisplay

} xfsPINCapabilitiesDisplayType;

/******************************************************************************
*                                                                             *
*       MIB Variables for the Status Table                                    *
*                                                                             *
******************************************************************************/
#define        xfsPINStatusManagedServiceName      (1)
#define        xfsPINStatusNumberSubDevices        (2)
#define        xfsPINStatusDevice                  (3)
#define        xfsPINStatusEncStat                 (4)
#define        xfsPINStatusCertificateState        (5)
#define        xfsPINStatusGuidancePinPad          (6)
```

```
#define        xfsPINStatusAutoBeepMode            (7)
#define        xfsPINStatusDevicePosition          (8)
#define        xfsPINStatusPowerSaveRecoveryTime   (9)

#define        xfsPINStatusExtraStatus             (100)


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

/*****************************************************************
*                                                                *
*  MIB Variables for the Capabilities Table                      *
*                                                                *
*****************************************************************/


#define        xfsPINCapabilitiesManagedServiceName         (1)
#define        xfsPINCapabilitiesDeviceType                 (2)
#define        xfsPINCapabilitiesCompoundDevice             (3)
#define        xfsPINCapabilitiesKeyNumber                  (4)
#define        xfsPINCapabilitiesAlgorithm                  (5)
#define        xfsPINCapabilitiesPinFormats                 (6)
#define        xfsPINCapabilitiesDerivationAlgorithms       (7)
#define        xfsPINCapabilitiesPresentationAlgorithms     (8)
#define        xfsPINCapabilitiesDisplay                    (9)
#define        xfsPINCapabilitiesConnect                    (10)
#define        xfsPINCapabilitiesIDKey                      (11)
#define        xfsPINCapabilitiesValidationAlgorithms       (12)
#define        xfsPINCapabilitiesKeyCheckModes              (13)
#define        xfsPINCapabilitiesGuidancePinPad             (14)
#define        xfsPINCapabilitiesPINCanPersistAfterUse      (15)
#define        xfsPINCapabilitiesAutoBeep                   (16)
#define        xfsPINCapabilitiesHSMVendor                  (17)
#define        xfsPINCapabilitiesHSMJournaling              (18)
#define        xfsPINCapabilitiesRSAAuthenticationScheme    (19)
#define        xfsPINCapabilitiesRSASignatureAlgorithm      (20)
#define        xfsPINCapabilitiesRSACryptAlgorithm          (21)
#define        xfsPINCapabilitiesRSAKeyCheckMode            (22)
#define        xfsPINCapabilitiesSignatureScheme            (23)
#define        xfsPINCapabilitiesEMVImportSchemes           (24)
#define        xfsPINCapabilitiesEMVHashAlgorithm           (25)
#define        xfsPINCapabilitiesKeyImportThroughParts      (26)
#define        xfsPINCapabilitiesENCIOProtocols             (27)
#define        xfsPINCapabilitiesTypeCombined               (28)
#define        xfsPINCapabilitiesSetPinblockDataRequired    (29)
#define        xfsPINCapabilitiesKeyBlockImportFormats      (30)
#define        xfsPINCapabilitiesPowerSaveControl           (31)

#define        xfsPINCapabilitiesExtraCapability            (100)

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

#endif  /* __inc_xfsmibpin__h */
```