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

WORKSHOP AGREEMENT

CWA 13937-2

August 2000

ICS 35.240.40

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part 2: Pin Keypad Device Class Interface - Programmer's Reference

Ref. No CWA 13937-2:2000 E

Rue de Stassart, 36 • B-1050 Bruxelles
Tel : +32 2 550 08 11 • Fax : +32 2 550 08 19

Foreword

This CWA contains the specifications that define the J/eXtensions for Financial Services (J/XFS) for the Java ™ Platform, as developed by the J/XFS Forum and endorsed by the CEN/ISSS J/XFS Workshop. J/XFS provides an API for Java applications which need to access financial devices. It is hardware independent and, by using 100% pure Java, also operating system independent.

The CEN/ISSS J/XFS Workshop gathers suppliers (among others the J/XFS Forum members), service providers as well as banks and other financial service companies. A list of companies participating in this Workshop and in support of this CWA is available from the CEN/ISSS Secretariat. The specification was agreed upon by the J/XFS Workshop Meeting of 1999-12-15/16 in Geneva and a subsequent electronic review by the Workshop participants, and the final version was sent to CEN for publication on 2000/06-21.

The specification is continuously reviewed and commented in the CEN/ISSS J/XFS Workshop. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this one. The information published in this CWA is furnished for informational purposes only. CEN/ISSS makes no warranty expressed or implied, with respect to this document. Updates of the specification will be available from the CEN/ISSS J/XFS Workshop public web pages pending their integration in a new version of the CWA (see: http://www.cenorm.be/isss/workshop/j-XFS/cwa-updates).

The J/XFS specifications are now further developed in the CEN/ISSS J/XFS Workshop. CEN/ISSS Workshops are open to all interested parties offering to contribute. Parties interested in participating should contact the CEN/ISSS Secretariat (isss@cenorm.be). To submit questions and comments for the J/XFS specifications, please contact the CEN/ISSS Secretariat (isss@cenorm.be) who will be forwarding them to the J/XFS Workshop.

Questions and comments can also be submitted to the members of the J/XFS Forum, who are all CEN/ISSS J/XFS Workshop members, through the J/XFS Forum web-site http:///www.jxfs.com

This CWA is composed of the following parts:

- Part 1: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Base Architecture - Programmer's Reference
- Part 2: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Pin Keypad Device Class Interface - Programmer's Reference
- Part 3: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Magnetic Stripe & Chip Card Device Class Interface - Programmer's Reference
- Part 4: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Text Input/Output Device Class Interface - Programmer's Reference
- Part 5: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Cash Dispenser, Recycler and ATM Interface - Programmer's Reference
- Part 6: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Printer Device Class Interface - Programmer's Reference
- Part 7: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Alarm Device - Programmer's Reference
- Part 8: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Sensors and Indicators Unit Device Class Interface - Programmer's Reference
- Part 9: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Depository Device Class Interface - Programmer's Reference
- Part 10: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Check Reader/Scanner Device Class Interface - Programmer's Reference

Contents

1 Scope

This document describes the Pin Keypad Device (PIN) classes based on the basic architecture of J/XFS which is similar to the JavaPOS architecture. It is event driven and asynchronous.

Three basic levels are defined in JavaPOS. For J/XFS this model is extended by a communication layer, which provides device communication that allows distribution of applications and devices within a network. So we have the following layers in J/XFS :

- Application
- Device Control and Device Manager
- Device Communication
- Device Service

Application developers program against control objects and the Device Manager which reside in the Device Control layer. This is the usual interface between applications and J/XFS devices. Device Control objects access the Device Manager to find an associated Device Service. Device Service objects provide the functionality to access the real device (i.e. like a device driver).
During application startup the Device Manager is responsible for locating the desired Device Service object and attaching this to the requesting Device Control object. Location and/or routing information for the Device Manager reside in a central repository.

To support Pin Keypad devices the basic Device Control structure is extended with various properties and methods specific to this device which are described on the following pages.

2 Overview

2.1 Description

This specification covers the interfaces and classes to access personal identification keypads (PIN pads). The main functions of PIN Keypad devices supported in this specification are:

- Non secure key pad functions (like key press detection, plain PIN retrieval,...)
- Secure PIN operations (like PIN validation, data encryption with PIN as crypotography input,...)
- Cryptographic services (like data encryption/decryption, MAC generation,...)

The J/XFS PIN Keypad specification separates the PIN Keypad functions between generic non-secure keypad functions and security-related functions, that is, the ones related to cryptography.

As well as the rest of J/XFS device controls, the J/XFS PIN Keypad Device Support uses the event driven model and the same behavioral model. Therefore, the application will instantiate a J/XFS PIN Keypad Device Control Object and then use the available methods to do I/O. When an I/O method is called, the J/XFS PIN Keypad Device Service will attempt to process the requested I/O. If the request is invalid or an exception is encountered, the application will be notified by a J/XFS exception. Completion of the request will be reported by an event. Thus the application must register itself with the J/XFS PIN Keypad Device Control Object for the various types of events it wishes to handle.

2.2 Classes and Interfaces

The following classes and interfaces are used by the J/XFS PIN Keypad Device Controls.

Class or Interface	Name	Description	Extends / Implements
Interface	**IJxfsBaseControl**	Base interface for all the device controls. Contains methods common to all the device controls.	--
Interface	**IJxfsPINKeypadControl**	Base interface for PIN controls. Contains methods declarations specific to PIN device controls.	Extends: **IJxfsBaseControl**
Interface	**IJxfsSecurePINKeypadControl**	Interface for PIN controls implementing secure PIN entry and validation. Contains methods specific to device controls for the secure PIN device category.	Extends: **IJxfsPINKeypadControl**
Interface	**IJxfsCrypto**	Interface for PIN controls implementing security and cryptographic functions.	Extends: **IJxfsPINKeypadControl**
Class	**JxfsBaseControl**	Base class for all the device controls. Contains properties common to all the device controls.	
Class	**JxfsPINKeypad**	Base class for PIN controls. Contains properties specific to PIN device controls.	Implements: **IJxfsPINKeypadControl**
Class	**JxfsSecurePINKeypad**	Class for PIN controls implementing security and cryptographic functions.	Extends: **JxfsPINKeypad** Implements: **IJxfsSecurePINKeypadControl, IJxfsCrypto**

2.3 Support Classes

Class or Interface	Name	Description	Extends / Implements
Interface	**JxfsConst**	Interface containing the Jxfs constants that are common to several device categories	--
Interface	**JxfsPINConst**	Interface containing the Jxfs constants that are common to all the PIN device controls.	--
Class	**JxfsPINFKeySet**	PIN function keys selector class. Indicates for each function key if it is selected or not. Properties are read only.	Extends: **JxfsType**
Class	**JxfsPINFKeysSelection**	Subclass of JxfsPINFKeySet. It contains the same properties, but they can be set by applications.	Extends: **JxfsPINFKeySet**
Class	**JxfsPINFDKeysSelection**	PIN function descriptor keys selector class. Indicates for each function descriptor key if it is selected or not.	Extends: **JxfsType**
Class	**JxfsPINFDKey**	Data class that contains information about a function descriptor key (FDKey).	Extends: **JxfsType**
Class	**JxfsPINReadMode**	Data class that defines the conditions for PIN keypad input operations.	Extends: **JxfsType**
Class	**JxfsPINPressedKey**	Data class that contains information about a key pressed during an input operation.	Extends: **JxfsType**
Class	**JxfsPINReadData**	Data class that contains the information provided to the application when an input operation completes.	Extends: **JxfsType**
Class	**JxfsPINFormats**	PIN formats selector class. Indicates for each PIN format if it is selected or not. Properties are read only.	Extends: **JxfsType**
Class	**JxfsPINValidationAlgorithms**	PIN validation algorithms selector class. Indicates for each PIN validation algorithm if it is selected or not. Properties are read only.	Extends: **JxfsType**
Class	**JxfsPINChipPresentationModes**	PIN chip presentation algorithms selector class. Indicates which presentation algorithms for chip PIN validation are supported.	Extends: **JxfsType**
Class	**JxfsPINValidationData**	Abstract data class. Root of a hierarchy of data objects that contain data for PIN verification and used in *validationPIN()* method.	Extends: **JxfsType**
Class	**JxfsPINValidationDataForDES**	Data class for PIN verification using DES	Extends: **JxfsPINValidationData**

		algorithm.	
Class	**JxfsPINValidationDataForEC**	Data class for PIN verification using EUROCHEQUE specification.	Extends: **JxfsPINValidationData**
Class	**JxfsPINValidationDataForVISA**	Data class for PIN verification using VISA specification.	Extends: **JxfsPINValidationData**
Class	**JxfsPINOffsetData**	Data class for creating a PIN offset.	Extends: **JxfsPINValidationData**
Class	**JxfsPINBlockData**	Data class for creating a PIN block.	Extends: **JxfsPINValidationData**
Class	**JxfsPINChipValidationData**	Abstract data class for all PIN chip validation modes.	Extends: **JxfsType**
Class	**JxfsPINChipValidationDataClear**	Data class for PIN chip validation mode Clear. Used as parameter in *validatePINChip* () method.	Extends: **JxfsPINChipValidationData**
Class	**JxfsPINValidationResult**	Data class that contains the result of a PIN validation operation.	Extends: **JxfsType**
Class	**JxfsPINOffset**	Data class that contains computed PIN offset.	Extends: **JxfsType**
Class	**JxfsPINBlock**	Data class that contains computed PIN block.	Extends: **JxfsType**
Class	**JxfsPINChipValidationResult**	Data class that contains the result of a PINchip validation operation.	Extends: **JxfsType**
Class	**JxfsPINCryptoModes**	Encyption modes selector class. Indicates for each encryption mode if it is selected or not. Properties are read only.	Extends: **JxfsType**
Class	**JxfsPINKeyDetail**	Data class containing information about a key from the device's key table.	Extends: **JxfsType**
Class	**JxfsPINKeyToImport**	Data class containing input data for *importKey()* method	Extends: **JxfsType**
Class	**JxfsPINInitialization**	Data class that contains result data from initialization of security module.	Extends: **JxfsType**
Class	**JxfsPINKeyVerificationData**	Data class that contains result data from an import key operation.	Extends: **JxfsType**
Class	**JxfsPINCryptoData**	Data class that contains input data for encrypt/decrypt operations.	Extends: **JxfsType**
Class	**JxfsPINMACData**	Data class that contains input data for MAC generation operation.	Extends: **JxfsPINCryptoData**
Class	**JxfsPINCryptoResult**	Data class that contains result data from cryptographic operations.	Extends: **JxfsType**
Class	**JxfsPINKeyUses**	Data class that contains	Extends:

		information on allowed uses for a key.	**JxfsType**
Class	**JxfsPINIdKeyModes**	Data class that contains information on implemented uses of ID key.	Extends: **JxfsType**
Class	**JxfsEvent**	Abstract class from which all Jxfs event classes are extended	Extends: **java.util. EventObject**
Class	*Event***Event**	The Device Service creates *Event* event instances of this class and delivers them through the J/XFS PIN Device Control's event callbacks to the application	Extends: **JxfsEvent**
Class	**JxfsException**	Exception class. The J/XFS PIN Device Control creates and throws exceptions on method failure and property access failure.	Extends: **java.lang.Exception**

3 Device behavior

3.1 Device open()

During the device open call the Device Service tries to access the connected device. This fails for the following circumstances:

JXFS_E_HARDWAREERROR	If the device could not be accessed. This may be that the device is not connected or broken.
JXFS_E_OPEN	The open was already done by this Device Control.

4 Classes and Interfaces

All operation methods return an identificationID. If a method cannot be processed, a JxfsException is thrown.
After processing has taken place, an OperationCompleteEvent is generated which contains detailed information about the status of the operation, i.e., if it failed or succeeded, and eventually additional data as a result.

The Constants, Error Codes, Exceptions, Status Codes and Support Classes that are used in the methods are described in special chapters at the end of the documentation.

4.1 Access to properties

Please note the following when determining the meaning of a property's **Access**:

R	The property is read only.
W	The property is write only.
R/W	The property may be read or written.

To access these properties the applications must use the appropriated methods specified by the JavaBean specification. Note that boolean properties are read using *isProperty* method instead of *getProperty*.

getProperty

Syntax	**Property *get*Property *() throws JxfsException***
Description	Returns the requested property.
Parameter	**None**
Event	No additional events are generated.
Exceptions	Some possible JxfsException *value codes*. Common values are: JXFS_E_CLOSED JXFS_E_UNREGISTERED JXFS_E_REMOTE

set*Property*

Syntax	**void *set*Property *(value) throws JxfsException***
Description	Sets the requested property.
Parameter	The desired property value.
Event	No additional events are generated
Exceptions	Some possible JxfsException *value codes*. Common values are: JXFS_E_CLOSED JXFS_E_UNREGISTERED JXFS_E_REMOTE JXFS_E_PARAMETER_INVALID

4.2 Exceptions

All the methods described for the specified interfaces can throw at least some of the following exceptions:

Value	**Meaning**
JXFS_E_CLOSED	The Device Control has not been opened.
JXFS_E_UNREGISTERED	The device is not registered at the JxfsDeviceManager.
JXFS_E_REMOTE	A network error ocurred.
JXFS_E_CLAIMED	The device is already claimed..
JXFS_E_PARAMETER_INVALID	A parameter is invalid.
JXFS_E_NOT_SUPPORTED	The function is not supported.

Only if a method can throw additional exceptions this is explicitly mentioned.

4.3 IJxfsPINKeypadControl

4.3.1 Introduction

The J/XFS PIN Keypad Device Control Subclass is defined in JxfsPINKeypad and is a subclass of JxfsBaseControl. . Its interface is defined in IJxfsPINKeypadControl interface which is a subclass of IJxfsBaseControl interface. The purpose of the J/XFS PIN Keypad Device Control object is to allow passing data and control between the application and the device support code so that the associated device can be accessed.

The JxfsPINKeypad class represents a physical PIN Keypad device with basic input keypad functions. There are no built-in security functions.

Summary

Although IJxfsPINKeypadControl is an interface, and therefore properties do not apply, properties are detailed here with the objective to provide guidance on the implementation of those classes that will implement this interface.

Therefore, the IJxfsPINKeypadControl consists on the following methods:

- Getters of listed properties.
- Methods listed.

Implements : **Extends :** *IJxfsBaseControl*

Property	Type	Access	Initialized after
supportedFDKeys	java.util.Vector	R	
supportedFKeys	**JxfsPINFKeySet**	R	
inputRawSupported	boolean	R	
inputCookedSupported	boolean	R	
beepOnPressSupported	boolean	R	

Method	Return	May use after
get*Property*	*Property*	
readData	identificationID	

4.3.2 Properties

supportedFDKeys Property (R)

Type	***java.util.Vector***
Initial Value	Depends on device.
Description	This vector contains a list of all function descriptor keys (FDKeys) supported by the device. Each vector element is a **JxfsPINFDKey** object that contains its key code and position information. See JxfsPINFDKey class description for more information. If empty, then no FDKeys are supported.

supportedFKeys Property (R)

Type	***JxfsPINFKeySet***
Initial Value	Null until open.
Description	Indicates the set of function keys supported by the device.

inputRawSupported (R)

Type	***boolean***
Initial Value	Depends on device.
Description	Specifies if raw input mode is supported by the device, where each key pressed during an input operation will generate an intermediate event. These events will contain information about pressed keys.

Value	**Meaning**
FALSE	Raw input mode is not supported.
TRUE	Raw input mode is supported.

inputCookedSupported (R)

Type	***boolean***
Initial Value	Depends on device.
Description	Specifies if cooked input mode is supported by the device, where no intermediate events per key pressed are generated. Data entered during an input operation is provided in the *OperationCompleteEvent* event.

Value	**Meaning**
FALSE	Cooked input mode is not supported.
TRUE	Cooked input mode is supported.

beepOnPressSupported (R)

Type	***boolean***
Initial Value	Depends on device.
Description	Specifies if the device has controllable capability of emitting an audible sound when a key is pressed.

Value	**Meaning**
FALSE	Device has no controllable beep capability.
TRUE	Device has controllable beep capability.

4.3.3 Methods

readData Method

Syntax ***identificationID readData (JxfsPINReadMode readMode) throws JxfsException;***

Description This command activates the PIN Keypad to read a data entry.

Digits are read until the value of *maxLength* property of *readMode* parameter is reached (if *autoEnd* property of *readMode* is set to TRUE), or a termination key is pressed. If *maxLength* is set to zero and no termination keys are specified, operation will not terminate until cancelled.

Each key pressed is notified as an intermediate event if *inputMode* property of *readMode* parameter is set to JXFS_PIN_INPUT_RAW. If *inputMode* is set to JXFS_PIN_INPUT_COOKED, then, a single *OperationCompleteEvent* event (containing input data) is issued when input operation terminates.

Parameter

Type	IO	Name	Meaning
JxfsPINReadMode	I	readMode	A data object that contains all the data required to perform a data entry (see *JxfsPINReadMode* class specification).

Event

OperationCompleteEvent
When an input operation is completed an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners

Field	Value
operationID	JXFS_O_PIN_READPIN
identificationID	Identification Id of complete operation.
result	
	JXFS_RC_SUCCESSFUL Operation completed successfully.
	JXFS_E_CANCELLED Operation was cancelled.
	JXFS_E_PIN_READ_FAILURE Read error.
data	A **JxfsPINReadData** object.

IntermediateEvent
Every key pressed generates an intermediate event if *inputMode* property is set to JXFS_PIN_INPUT_RAW.
IntermediateEvent events are sent by PIN Device Control to all registered IntermediateListeners

Field	Value
operationID	JXFS_O_PIN_READPIN
identificationID	Identification Id of operation.
reason:	
	JXFS_I_PIN_KEY_PRESSED A key has been pressed.
data	A **JxfsPINPressedKey** object.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_KEYINVALID	At least one of the specified active function keys or FDKeys is invalid.
JXFS_E_PIN_NOACTIVEKEYS	No active function key or FDKey specified.

JXFS_E_PIN_KEYNOTSUPPORTED	At least one of the specified active function keys or FDKeys (*activeFKeys* or *activeFDKeys* properties of *readMode* parameter) is not supported by the device service.
JXFS_E_PIN_MINIMUNLENGTH	The *minLength* property is invalid or greater than the *maxLength* property.

4.4 IJxfsSecurePINKeypadControl

4.4.1 Introduction

The J/XFS Secure PIN Keypad Device Control Subclass is defined in JXFSecurePINKeypad and is a subclass of JxfsPINKeypad. The Secure PIN Keypad Device Control is intended to match physical PIN Keypad devices with the following extended security capabilities:

- PIN secure read,
- PIN verification and
- Cryptographic services.

Its interface is defined in IJxfsSecurePINKeypadControl interface which is a subclass of IJxfsPINKeypadControl interface.

Summary

Although IJxfsSecurePINKeypadControl is an interface, and therefore properties do not apply, properties are detailed here with the objective to provide guidance on the implementation of those classes that will implement this interface.

Therefore, the IJxfsSecurePINKeypadControl consists on the following methods:

- Getters of listed properties.
- Methods listed.

Implements : **Extends :** *IJxfsPINKeypadControl*

Property	Type	Access	Initialized after
supportedPINFormats	**JxfsPINFormats**	R	
supportedValidation Algorithms	**JxfsPINValidationAlgorithms**	R	
supportedChipPresentation Modes	**JxfsPINChipPresentationModes**	R	

Method	Return	May use after
get*Property*	*Property*	
secureReadPIN	identificationID	
createOffset	identificationID	
createPINBlock	identificationID	
validatePIN	identificationID	
createOffsetSecure	identificationID	
createPINBlockSecure	identificationID	
validatePINSecure	identificationID	
validatePINChip	identificationID	

4.4.2 Properties

supportedPINFormats Property (R)

Type	***JxfsPINFormats***
Initial Value	Null until open.
Description	Specifies the supported PIN formats.

supportedValidationAlgorithms Property (R)

Type	***JxfsPINValidationAlgorithms***
Initial Value	Null until open.
Description	Specifies the supported algorithms for PIN validation.

supportedChipPresentationModes Property (R)

Type	***JxfsPINChipPresentationModes***
Initial Value	Depends on device.
Description	Specifies the supported presentation algorithms for chip PIN validation.

4.4.3 Methods

secureReadPIN Method

Syntax ***identificationID secureReadPIN (JxfsPINReadMode readMode) throws JxfsException;***

Description This command activates the PIN Keypad to read a PIN entry in a secure way.
Entered data is not passed to the application but retained for further cryptographic operation (like PIN validation, PIN offset generation or PIN Block generation).

Digits are read until the value of *maxLength* property of *readMode* parameter is reached (if *autoEnd* property of *readMode* is set to TRUE), or a termination key is pressed. If *maxLength* is set to zero and no termination keys are specified, operation will not terminate until cancelled.

Each key pressed is notified as an intermediate event if *inputMode* property of *readMode* parameter is set to JXFS_PIN_INPUT_RAW. If *inputMode* is set to JXFS_PIN_INPUT_COOKED, then, a single *OperationCompleteEvent* event (containing input data) is issued when input operation terminates.

Parameter

Type	IO	Name	Meaning
JxfsPINReadMode	I	readMode	A data object that contains all the data required to perform a data entry (see *JxfsPINReadMode* class specification).

Event

OperationCompleteEvent
When an input operation is completed an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners

Field	Value
operationID	JXFS_O_PIN_READPIN
identificationID	Identification Id of complete operation.
result	
	JXFS_RC_SUCCESSFUL Operation completed successfully.
	JXFS_E_CANCELLED Operation was cancelled.
	JXFS_E_PIN_READ_FAILURE Read error.
data	A **JxfsPINReadData** object.

IntermediateEvent
Every key pressed generates an intermediate event if *inputMode* property is set to JXFS_PIN_INPUT_RAW.
IntermediateEvent events are sent by PIN Device Control to all registered IntermediateListeners

Field	Value
operationID	JXFS_O_PIN_READPIN
identificationID	Identification Id of operation.
reason:	
	JXFS_I_PIN_KEY_PRESSED A key has been pressed.
data	A **JxfsPINPressedKey** object.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning

JXFS_E_PIN_KEYINVALID	At least one of the specified active function keys or FDKeys is invalid.
JXFS_E_PIN_NOACTIVEKEYS	No active function key or FDKey specified.
JXFS_E_PIN_KEYNOTSUPPORTED	At least one of the specified active function keys or FDKeys (*activeFKeys* or *activeFDKeys* properties of *readMode* parameter) is not supported by the device service.
JXFS_E_PIN_MINIMUNLENGTH	The *minLength* property is invalid or greater than the *maxLength* property.

createOffset Method

Syntax ***identificationID createOffset (JxfsOffsetData offsetData) throws JxfsException;***

Description This function is used to generate a PIN Offset that is used to verify PINs using the *validatePIN()* method with DES validation algorithm.

The PIN offset is computed by combining validation data with the keypad entered PIN.

This method clears the PIN.

Parameter

Type	IO	Name	Meaning
JxfsOffsetData	I	offsetData	A data object that contains all the data required to create the PIN offset (see *JxfsOffsetData* class specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners. In addtion a data object is returned:

Field	Value
operationID	JXFS_O_PIN_CREATEOFFSET
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.

	data	A **JxfsPINOffset** object. It contains the computed PIN offset

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_NOTSUPPORTEDCAP	Offset generation is not supported.

createPINBlock Method

Syntax ***identificationID createPINBlock (JxfsPINBlockData pinBlockData) throws JxfsException;***

Description This method takes the account information and a PIN entered by the user to build a formatted PIN. Encrypting this formatted PIN once or twice returns a PIN block which can be written on a magnetic card or sent to a host.

The PIN block can be calculated using one of the formats specified in the *supportedPINFormats* property.

The PIN block is computed by combining customer data with the keypad entered PIN.

This command clears the PIN.

Parameter

Type	IO	Name	Meaning
JxfsPINBlockData	I	pinBlockData	A data object that contains all the data required to create the PIN block (see *JxfsPINBlockData* class specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners.

Field	Value
operationID	JXFS_O_PIN_CREATEPINBLOCK
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN_NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.

	data	A **JxfsPINBlock** object. It contains the computed PIN block.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_FORMAT_NOTSUPPORTED	The specified PIN block format is not supported.

validatePIN Method

Syntax ***identificationID validatePIN (JxfsPINValidationData validationData) throws JxfsException;***

Description The previously entered PIN is combined with the requisite data specified by the PIN validation algorithm and locally verified for correctness.

The validationData object should specify the validation algorithm to be used for PIN validation as well as all needed data to perform the validation (*see JxfsPINValidationData* class specification)

This method clears the PIN.

Parameter

Type	IO	Name	Meaning
JxfsPINValidationData	I	validationData	Validation data object containing specific data for the actual PIN validation algorithm to be used *(see JxfsPINValidationData* class specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners.

Field	Value
operationID	JXFS_O_PIN_VALIDATEPIN
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN_NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.

	data	A **JxfsPINValidationResult** object. It contains the results of the validation.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_NOTSUPPORTEDCAP	The requested validation algorithm is not supported.

createOffsetSecure Method

Syntax ***identificationID createOffsetSecure (JxfsPINOffsetData offsetData) throws JxfsException;***

Description This function is used to generate a PIN Offset that is used to verify PINs using the *validatePIN()* method with DES validation algorithm.

With combined MSD-PIN devices, this funcion does not require that validation data data be first read from the card with the MSD component and then returned to the device as a parameter. Instead, the validation data is automatically read from the card in the device.
The behavior is as follows:
1 – If card is present in reader and ejectCurrent property is false then go to 5.
2 – If card is present in reader and ejectCurrent property is true then eject the card.
3 – Arm the device to accept a magnetic stripe card.
4 – Poll card status and verify that card is seated.
5 – Perform the intended function using the offset data read from the card.
6 – Eject the card if ejectWhenComplete property is true.

This method clears the PIN.

Parameter

Type	IO	Name	Meaning
JxfsPINOffsetData	I	offsetData	A data object that contains all the data required to create the PIN offset (see *JxfsOffsetData* class specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners. In addtion a data object is returned:

Field	Value
operationID	JXFS_O_PIN_CREATEOFFSET_SECURE
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found.

		JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason. JXFS_E_MSD_READFAILURE No read conditions were satisfied JXFS_E_MSD_NOMEDIA Media was removed before operation completion. JXFS_E_MSD_INVALIDMEDIA No appropriated media was found. JXFS_E_MSD_MEDIAJAMMED Media is jammed. JXFS_E_MSD_SHUTTERFAIL Shutter could not be opened.
	data	A **JxfsPINOffset** object. It contains the computed PIN offset

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_NOTSUPPORTEDCAP	Secure offset generation is not supported.
JXFS_E_MSD_NOTSUPPORTEDTRACK	Track specified in *validationTrackNumber* property is not supported by the device.

createPINBlockSecure Method

Syntax ***identificationID createPINBlockSecure (JxfsPINBlockData pinBlockData) throws JxfsException;***

Description This method takes the account information and a PIN entered by the user to build a formatted PIN. Encrypting this formatted PIN once or twice returns a PIN block which can be written on a magnetic card or sent to a host.

The PIN block can be calculated using one of the formats specified in the *supportedPINFormats* property.

The PIN block is computed by combining customer data with the keypad entered PIN.

With combined MSD-PIN devices, this funcion does not require that customer data be returned to the device as a parameter. Instead, the customer data is automatically read from the card in the device.
The behavior is as follows:

1 – If card is present in reader and ejectCurrent property is false then go to 5.
2 – If card is present in reader and ejectCurrent property is true then eject the card.
3 – Arm the device to accept a magnetic stripe card.
4 – Poll card status and verify that card is seated.
5 – Perform the intended function using the customer data read from the card.

6 – Eject the card if ejectWhenComplete property is true.

This command clears the PIN.

Parameter

Type	IO	Name	Meaning
JxfsPINBlockData	I	pinBlockData	A data object that contains all the data required to create the PIN block (see *JxfsPINBlockData* class specification).

Event

OperationCompleteEvent
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners.

Field	Value
operationID	JXFS_O_PIN_CREATEPINBLOCK_SECURE
identificationID	Identification Id of complete operation.
result	
	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN_NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason. JXFS_E_MSD_READFAILURE No read conditions were satisfied JXFS_E_MSD_NOMEDIA Media was removed before operation completion. JXFS_E_MSD_INVALIDMEDIA No appropriated media was found. JXFS_E_MSD_MEDIAJAMMED Media is jammed. JXFS_E_MSD_SHUTTERFAIL Shutter could not be opened.
data	A **JxfsPINBlock** object. It contains the computed PIN block.

Exceptions

Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_NOTSUPPORTEDCAP	Secure block generation is not supported.
JXFS_E_PIN_FORMAT_NOTSUPPORTED	The specified PIN block format is not supported.

JXFS_E_MSD_NOTSUPPORTEDTRACK	Track specified in *validationTrackNumber* property is not supported by the device.

validatePINSecure Method

Syntax ***identificationID validatePINSecure (JxfsPINValidationData validationData) throws JxfsException;***

Description The previously entered PIN is combined with the requisite data specified by the DES validation algorithm and locally verified for correctness.

With combined MSD-PIN devices, this funcion does not require that offset and/or validation data be returned to the device as parameters. Instead, offset and/or validation data can be automatically read from the card in the device.
The behavior is as follows:
1 – If card is present in reader and ejectCurrent property is false then go to 5.
2 – If card is present in reader and ejectCurrent property is true then eject the card.
3 – Arm the device to accept a magnetic stripe card.
4 – Poll card status and verify that card is seated.
5 – Perform the intended function using the data read from the card.
6 – Eject the card if ejectWhenComplete property is true.

This method clears the PIN.

Parameter

Type	IO	Name	Meaning
JxfsPINValidationData	I	validationData	Validation data object containing specific data for the actual PIN validation algorithm to be used *(see JxfsPINValidationData* class specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteListeners.

Field	Value
operationID	JXFS_O_PIN_VALIDATEPIN_SECURE
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN_NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded.

		JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason. JXFS_E_MSD_READFAILURE No read conditions were satisfied JXFS_E_MSD_NOMEDIA Media was removed before operation completion. JXFS_E_MSD_INVALIDMEDIA No appropriated media was found. JXFS_E_MSD_MEDIAJAMMED Media is jammed. JXFS_E_MSD_SHUTTERFAIL Shutter could not be opened.
	data	A **JxfsPINValidationResult** object. It contains the results of the validation.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_NOTSUPPORTEDCAP	Secure PIN validation is not supported.
JXFS_E_MSD_NOTSUPPORTEDTRACK	Tracks specified in *validationTrackNumber* and/or *offsetTrackNumber* properties are not supported by the device.

validatePINChip Method

Syntax ***identificationID validatePINChip (java.lang.String aCCDeviceName, JxfsPINChipValidationData validationData) throws JxfsException;***

Description The previously entered PIN is combined with the requisite data specified by the chip PIN presentation algorithm and presented to the chip card device for correctness verification.

The validationData object specifies all the needed data to perform the validation (*see JxfsPINChipValidationData* class and subclasses specifications)

This method clears the PIN.

Parameter	Type	IO	Name	Meaning

	java.lang.String	I	aCCDeviceName	The name of the Chip Card device to be used for PIN validation. It is responsibility of the application to ensure the chip card has already been inserted. It is responsibility of the J/XFS device service to instantiate a J/XFS Chip Card Control and to use it exclusively to access the chip card. If the chip card device is already claimed by someone else, a JXFS_E_CLAIMED exception is thrown. The device service must release ownership of the device after using it. During the validation of the PIN the application must not access the chip card; only the PinPad device service has the right to access the chip card.
	JxfsPINChipValidationData	I	validationData	Validation data object containing specific data for the actual PIN validation algorithm to be used *(see JxfsPINChipValidationData* class specification).

Event **OperationCompleteEvent**

When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteEvent listeners.

Field	**Value**
operationID	JXFS_O_PIN_VALIDATEPINCHIP
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_NO_PIN PIN has not been entered or has been cleared. JXFS_E_PIN_NOT_ALLOWED PIN entered by the user is not allowed. JXFS_E_CCD_IOERROR IO error occurred. No data is available. Verification could not be performed. JXFS_E_CCD_NOMEDIA Media was removed before operation completion JXFS_E_CCD_INVALIDMEDIA No appropriated media was found. JXFS_E_CCD_MEDIAJAMMED Media is jammed. JXFS_E_CCD_SHUTTERFAIL Shutter could not be opened.

		JXFS_E_CCD_BADDATA Chip reported data was bad. JXFS_E_CCD_BADPROTOCOL Protocol not supported.
	data	A **JxfsPINChipValidationResult** object. It contains the results of the validation.

IntermediateEvent

IntermediateEvent events can be sent by PIN Device Control to all registered IntermediateListeners

Field	**Value**
operationID	JXFS_O_CCD_CHIPIO
identificationID	Identification Id of operation.
reason:	
	JXFS_I_CCD_NO_MEDIA_PRESENT The read operation request cannot progress because there is no media inserted. JXFS_I_CCD_MEDIA_INSERTED The read operation request continues because a media has been inserted.
data	null

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PIN_CHIPPRES_NOTSUPPORTED	The requested chip presentation algorithm is not supported.

4.5 IJxfsCrypto

4.5.1 Introduction

The cryptographic services interface provides generic cryptography functions. It handles a key table and allows the user to encrypt, decrypt or calculate check codes using keys from its table. This interface is used for the sake of clarity, to separate the generic cryptographic functions from the PIN related cryptographic functions. The JxfsSecurePINKeypad class implements this interface.

Summary

Implements : - **Extends :** *IJxfsPINKeypadControl*

Property	**Type**	**Access**	**Initialized after**
supportedCryptoModes	**JxfsPINCryptoModes**	R	
numberOfKeys	int	R	
idKey	**JxfsPINIdKeyModes**	R	

Method	**Return**	**May use after**
get*Property*	*Property*	
decrypt	identificationID	
encrypt	identificationID	
generateMAC	identificationID	
getKeyInfo	**JxfsPINKeyDetail**	
getKeyNameList	java.util.Vector	
importKey	identificationID	
initialize	identificationID	

4.5.2 Properties

supportedCryptModes Property (R)

Type	***JxfsPINCryptoModes***
Initial Value	Depends on device.
Description	Specifies the supported encryption modes.

numberOfKeys Property (R)

Type	***int***
Initial Value	Depends on device.
Description	Specifies the number of keys that may be stored by the device.

idKey Property (R)

Type	***JxfsPINIdKeyModes***
Initial Value	Depends on device.
Description	Specifies whether an ID key is supported or not.

4.5.3 Methods

decrypt Method

Syntax ***identificationID decrypt (JxfsPINCryptoData decryptData) throws JxfsException;***

Description Deciphers data with the currently selected algorithm and the specified key name.

Parameter

Type	IO	Name	Meaning
JxfsPINCryptoData	I	decryptData	Contains the data and additional information required to perform a decrypt operation. See *JxfsPINCryptoData* specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteEvent listeners.

Field	Value
operationID	JXFS_O_PIN_DECRYPT
identificationID	Identification Id of complete operation.
result	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key JXFS_E_PIN_LENGTH_ERROR The length of the start value specified is not supported. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.
data	A **JxfsPINCryptoResult** object. It contains the results of the decryption.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_CRYPTNOTSUPPORTED	The decryption method is not supported.

encrypt Method

Syntax ***identificationID encrypt (JxfsPINCryptoData encryptData) throws JxfsException;***

Description Encrypts data with the currently selected algorithm and the specified key name.

Parameter

Type	IO	Name	Meaning

	JxfsPINCryptoData	I	encryptData	Contains the data and additional information required to perform a encrypt operation. See *JxfsPINCryptoData* specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteEvent listeners.

Field	**Value**
operationID	JXFS_O_PIN_ENCRYPT
identificationID	Identification Id of complete operation.
result	
	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key JXFS_E_PIN_LENGTH_ERROR The length of the start value specified is not supported. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.
data	A **JxfsPINCryptoResult** object. It contains the results of the encryption.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PIN_CRYPTNOTSUPPORTED	The encryption method is not supported.

generateMAC Method

Syntax ***identificationID generateMAC (JxfsPINMACData macData) throws JxfsException;***

Description Generates a MAC data with the currently selected algorithm.

Parameter	**Type**	**IO**	**Name**	**Meaning**
	JxfsPINMACData	I	macData	Contains the data and additional information required to perform a decrypt operation. See *JxfsPINMACData* specification).

Event **OperationCompleteEvent**
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteEvent listeners.

Field	**Value**
operationID	JXFS_O_PIN_GENMAC
identificationID	Identification Id of complete operation.
result	

		JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_KEY_NOT_FOUND The specified key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by this key JXFS_E_PIN_LENGTH_ERROR The length of the start value specified is not supported. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.
	data	A **JxfsPINCryptoResult** object. It contains the generated MAC.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_CRYPTNOTSUPPORTED	The encryption method is not supported.

getKeyInfo Method

Syntax ***JxfsPINKeyDetail getKeyInfo (java.lang.String keyName) throws JxfsException;***

Description Retrieves information about a given key
Returns a *JxfsPINKeyDetail* object with the requested info.

Parameter

Type	IO	Name	Meaning
String	I	keyName	Name of the key to be queried.

Event No additional events are generated:

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PIN_KEY_NOT_FOUND	The specified key was not found.

getKeyNameList Method

Syntax ***java.util.Vector getKeyNameList () throws JxfsException;***

Description Retrieves the list of keys names used by the device.
Returns a vector of strings with the name of all keys stored in the device.

Event No additional events are generated.

Exceptions No additional exceptions are generated.

importKey Method

Syntax ***identificationID importKey (JxfsPINKeyToImport keyToImport, boolean lastOrOnlyPart) throws JxfsException;***

Description Loads a key or part of a key into the encryption module. The key can be passed in clear text mode or encrypted with an accompanying “key encryption key”.

The imported key is imported into the encryption module and is used for cryptographic operations.

The key may be loaded in parts.

Parameter

Type	IO	Name	Meaning
JxfsPINKeyToImport	I	keyToImport	Contains the data required to import the key (see *JxfsPINKeyToImport* specification).
boolean	I	lastOrOnlyPart	If true, key import is finished.

Event

OperationCompleteEvent
When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationCompleteEvent listeners.

Field	Value
operationID	JXFS_O_PIN_IMPORTKEY
identificationID	Identification Id of complete operation.
result	
	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_KEY_NOT_FOUND The specified key encryption key was not found. JXFS_E_PIN_KEY_NO_VALUE The specified key is not loaded. JXFS_E_PIN_USE_VIOLATION The specified use is not supported by the specified key. JXFS_E_PIN_DUPLICATE_KEY A key exists with the specified name and cannot be overwritten. JXFS_E_PIN_LENGTH_ERROR The length of the key value specified is not supported. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.
data	A **JxfsPINKeyVerificationData** Object.

Status Event
If the completion of this operation results in an updated key in device's table key, then the J/XFS PIN Keypad device control will fire a StatusEvent to al registered listeners:

Field	Value
status	JXFS_S_PIN_KEY A new key has been loaded/imported into the device's key table.
details	A **JxfsPINKeyDetail** object containing information about the added key.

Exceptions

No additional exceptions are generated.

initialize Method

Syntax ***identificationID initialize (byte[ ] id, byte[ ] key) throws JxfsException;***

Description Clears all loaded or imported keys from device's key table.

Usually this operation is invoked by an operator task and not by the application program.

During initialization, an optional encrypted Id key can be stored in the device. The Id key and the corresponding encryption key can be passed as paramenters; if not, they are generated automatically by the encryption module. The encrypyted Id is returned to the application and serves, if supported (see idKey property), as authorization for the key import function.

Parameter

Type	IO	Name	Meaning
byte[ ]	I	id	ID Key. This byte array is encrypted under *key* and stored into the device. **Null** if not required.
byte[ ]	I	key	Encryption key of *id*. It is also stored into the device. If **null**, *id* is in clear mode.

Event **OperationCompleteEvent**

When the operation completes an *OperationCompleteEvent* event will be sent by J/XFS PINKeypad Device Control to all registered OperationComplete event listeners.

Field	Value
operationID	JXFS_O_PIN_INITIALIZE
identificationID	Identification Id of complete operation.
result	
	JXFS_RC_SUCCESSFUL Operation completed successfully. JXFS_E_CANCELLED Operation was cancelled. JXFS_E_PIN_ACCESS_DENIED The encryption module is either not initialized or not ready for any vendor specific reason.
data	A **JxfsPINInitialization** Object.

Exceptions No additional exceptions are generated.

5 Support Classes

5.1 JxfsPINFKeySet

This class provides properties and methods to query which function keys are supported or are active.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
fk0	boolean	R	
fk1	boolean	R	
fk2	boolean	R	
fk3	boolean	R	
fk4	boolean	R	
fk5	boolean	R	
fk6	boolean	R	
fk7	boolean	R	
fk8	boolean	R	
fk9	boolean	R	
fkEnter	boolean	R	
fkCancel	boolean	R	
fkClear	boolean	R	
fkBackspace	boolean	R	
fkHelp	boolean	R	
fkDecPoint	boolean	R	
fk00	boolean	R	
fk000	boolean	R	

Method	Return	May use after
is*Property*	*Property*	
allFKeys	boolean	
noFKeys	boolean	
JxfsPINFKeySet	(constructor of the class)	

5.1.1 Properties

fk0 .. fk000 Properties (R)

Type ***boolean***
Initial Value FALSE
Description Indicates if related function key is selected.
Note: fk00 and fk000 (hundred's and thousand's keys) are treated as sequences of two and three fk0, respectively.

Value	Meaning
FALSE	Function key is not selected.
TRUE	Function key is selected.

5.1.2 Methods

allFKeys Method

Syntax ***boolean allFKeys ()***
Description Returns TRUE if all properties are set to TRUE.

noFKeys Method

Syntax ***boolean noFKeys ()***
Description Returns TRUE if all properties are set to FALSE.

JxfsPINFKeySet Constructor

Syntax ***JxfsPINFKeySet (boolean fk0, boolean fk1, ... , boolean fk000)***
Description Constructor of the class.

5.2 JxfsPINFKeysSelection

This class provides properties and methods to query and select which function keys are active.

Summary

Implements : **Extends :** **JxfsPINFKeySet**

Property	Type	Access	Initialized after
No additional properties.			

Method	Return	May use after
set*Property*	void	
setAllFKeys	void	
setNoFKeys	void	
JxfsPINFKeysSelection	(constructor of the class)	

5.2.1 Properties

No additional properties to those inherited from base class *JxfsPINFKeySet*.

5.2.2 Methods

setAllFKeys Method

Syntax ***void setAllFKeys ()***
Description Sets all properties to TRUE.

setNoFKeys Method

Syntax ***void setNoFKeys ()***
Description Sets all properties to FALSE.

JxfsPINFKeysSelection Constructor

Syntax ***JxfsPINFKeysSelection (boolean fk0, ... , boolean fk000)***
Description Constructor of the class.

5.3 JxfsPINFDKeysSelection

This class provides properties and methods to query and select which function descriptor keys (FDKeys) are active.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
fdk01	boolean	R/W	
fdk02	boolean	R/W	
fdk03	boolean	R/W	
fdk04	boolean	R/W	
fdk05	boolean	R/W	
fdk06	boolean	R/W	
fdk07	boolean	R/W	
fdk08	boolean	R/W	
fdk09	boolean	R/W	
fdk10	boolean	R/W	
fdk11	boolean	R/W	
fdk12	boolean	R/W	
fdk13	boolean	R/W	
fdk14	boolean	R/W	
fdk15	boolean	R/W	
fdk16	boolean	R/W	
fdk17	boolean	R/W	
fdk18	boolean	R/W	
fdk19	boolean	R/W	
fdk20	boolean	R/W	
fdk21	boolean	R/W	
fdk22	boolean	R/W	
fdk23	boolean	R/W	
fdk24	boolean	R/W	
fdk25	boolean	R/W	
fdk26	boolean	R/W	
fdk27	boolean	R/W	
fdk28	boolean	R/W	
fdk29	boolean	R/W	
fdk30	boolean	R/W	
fdk31	boolean	R/W	
fdk32	boolean	R/W	

Method	Return	May use after
is*Property*	*Property*	
set*Property*	void	
allFDKeys	boolean	
noFDKeys	boolean	
setAllFDKeys	void	
setNoFDKeys	void	
JxfsPINFDKeysSelection	(constructor of the class)	

5.3.1 Properties

fdk01 .. fdk32 Properties (R/W)

Type ***boolean***
Initial Value FALSE
Description Indicates if related function descriptor key is selected.

Value	Meaning
FALSE	Function descriptor key is not selected.
TRUE	Function descriptor key is selected.

5.3.2 Methods

allFDKeys Method

Syntax ***boolean allFDKeys ()***
Description Returns TRUE if all properties are set to TRUE.

noFDKeys Method

Syntax ***boolean noFDKeys ()***
Description Returns TRUE if all properties are set to FALSE.

setAllFDKeys Method

Syntax ***void setAllFDKeys ()***
Description Sets all properties to TRUE.

setNoFDKeys Method

Syntax ***void setNoFDKeys ()***
Description Sets all properties to FALSE.

JxfsPINFDKeysSelection Constructor

Syntax ***JxfsPINFDKeysSelection (boolean fdk01, ... , boolean fdk32)***
Description Constructor of the class.

5.4 JxfsPINFDKey

Summary

The JxfsPINFDKey class contains information about a function descriptor key (FDKey).

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
keyCode	int	R	
relativeX	int	R	
relativeY	int	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINFDKey	(constructor of the class)	

5.4.1 Properties

keyCode Property (R)

Type ***int***

Description Specifies the code used for the function descriptor key FDKey.
Its value is one of the following:

Value
JXFS_PIN_FK_FDK01
JXFS_PIN_FK_FDK02
JXFS_PIN_FK_FDK03
JXFS_PIN_FK_FDK04
JXFS_PIN_FK_FDK05
JXFS_PIN_FK_FDK06
JXFS_PIN_FK_FDK07
JXFS_PIN_FK_FDK08
JXFS_PIN_FK_FDK09
JXFS_PIN_FK_FDK10
JXFS_PIN_FK_FDK11
JXFS_PIN_FK_FDK12
JXFS_PIN_FK_FDK13
JXFS_PIN_FK_FDK14
JXFS_PIN_FK_FDK15
JXFS_PIN_FK_FDK16
JXFS_PIN_FK_FDK17
JXFS_PIN_FK_FDK18
JXFS_PIN_FK_FDK19
JXFS_PIN_FK_FDK20
JXFS_PIN_FK_FDK21
JXFS_PIN_FK_FDK22
JXFS_PIN_FK_FDK23
JXFS_PIN_FK_FDK24
JXFS_PIN_FK_FDK25
JXFS_PIN_FK_FDK26
JXFS_PIN_FK_FDK27
JXFS_PIN_FK_FDK28
JXFS_PIN_FK_FDK29
JXFS_PIN_FK_FDK30
JXFS_PIN_FK_FDK31
JXFS_PIN_FK_FDK32

relativeX Property (R)

Type ***int***

Description Specifies the FDKey position relative to the left hand side of the screen expressed as a percentage of the width of the screen.

relativeY Property (R)

Type	***int***
Description	Specifies the FDKey position relative to the top of the screen expressed as a percentage of the height of the screen.

5.4.2 Methods

JxfsPINFDKey Constructor

Syntax	***JxfsPINFDKey (int keyCode, int relativeX, int relativeY)***
Description	Constructor of the class.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PARAMETER_INVALID	Some parameter is out of range.

5.5 JxfsPINReadMode

This class specifies the conditions for PIN keypad data entry when using *readData()* and *secureReadPIN()* methods.

Summary

Implements : **Extends :** *JxfsType*

Property	Type	Access	Initialized after
activeFDKeys	**JxfsPINFDKeysSelection**	R/W	
activeFKeys	**JxfsPINFKeysSelection**	R/W	
terminateFDKeys	**JxfsPINFDKeysSelection**	R/W	
terminateFKeys	**JxfsPINFKeysSelection**	R/W	
autoEnd	boolean	R/W	
beepOnPress	boolean	R/W	
inputMode	int	R/W	
maxLength	int	R/W	
minLength	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINReadMode	(constructor of the class)	

5.5.1 Properties

activeFDKeys Property (R/W)

Type ***JxfsPINFDKeysSelection***
Initial Value Null until open.
Description Indicates the set of function descriptor keys (FDKeys) enabled for subsequent input operations.

activeFKeys Property (R/W)

Type ***JxfsPINFKeysSelection***
Initial Value Null until open.
Description Indicates the set of function keys enabled for subsequent input operations.

terminateFDKeys Property (R/W)

Type ***JxfsPINFDKeysSelection***
Initial Value Null until open.
Description Specifies the set of function descriptor keys (FDKeys) that, if pressed during an input operation, will terminate a data entry.
It must be a subset of the set defined by *activeFDKeys.*

terminateFKeys Property (R/W)

Type ***JxfsPINFKeysSelection***
Initial Value Null until open.
Description Specifies the set of function keys that, if pressed during an input operation, will terminate a data entry.

It must be a subset of the set defined by *activeFKeys.*

autoEnd Property (R/W)

Type ***boolean***
Initial Value FALSE
Description Indicates the criteria used to terminate subsequent input operations.

If *maxLength* is set to 0, this property is ignored and input is only terminated by a termination key (see *terminateFKeys* and *terminateFDKeys* properties).

Value	Meaning
TRUE	PIN entry terminates when the maximun number of digits are entered (*maxLength* property).
FALSE	PIN entry terminates when a termination key (*terminateFKey*s and *terminateFDKeys* properties) has been pressed. In this case, when *maxLength* is reached, numeric keys are disabled by the device service.

beepOnPress Property (R/W)

Type ***boolean***
Initial Value FALSE
Description Specifies if the device must generate an audible sound at every key press or not.

Value	Meaning
FALSE	The device must not beep.
TRUE	The device must beep.

inputMode Property (R/W)

Type ***int***
Initial Value JXFS_PIN_INPUT_COOKED
Description Specifies the input mode to be used in subsequent input operations.

Value	Meaning
JXFS_PIN_INPUT_RAW	Each key pressed during an input operation will generate an intermediate event. These events will contain information about pressed keys.
JXFS_PIN_INPUT_COOKED	No intermediate events per key pressed are generated. Data entered during an input operation is provided in the *OperationCompleteEvent* event.

maxLength Property (R/W)

Type ***int***
Initial Value 8
Description Specifies the maximum number of digits which can be entered in an input operation.

If autoEnd is set to TRUE, the input operation ends when this maximun number of digits has been entered.

If it is set to zero, the input operation does not ends until a termination key is pressed (see *terminateKeys* and *terminateFDKeys* properties). If no termination keys are specified, the input operation will not terminate until a *cancel()* operation is issued.

minLength Property (R/W)

Type	***int***
Initial Value	1
Description	Specifies the minimum number of digits which must be entered for a valid input operation. A value of JXFS_PIN_NO_MINUMUM_LENGTH (zero) indicates no minimum PIN length verification.

5.5.2 Methods

JxfsPINReadMode Constructor

Syntax	***JxfsPINReadMode (JxfsPINFDKeysSelection activeFDKeys, JxfsPINFKeysSelection activeFKeys, JxfsPINFDKeysSelection terminateFDKeys, JxfsPINFKeysSelection terminateFKeys, boolean autoEnd, boolean beepOnPress, int inputMode, int maxLength, int minLength)***
Description	Constructor of the class.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PARAMETER_INVALID	Any of the following conditions is met: *activeFDKeys* is null. *activeFKeys* is null. *terminateFDKeys* is null. *terminateFKeys* is null. *inputMode* is not one of the listed values. *maxLength* is less than *minLength.* *minLength* is negative.

5.6 JxfsPINPressedKey

This class contains the data associated to a pressed key during an input operation.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
keyCode	int	R	
keyType	int	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINPressedKey	(constructor of the class)	

5.6.1 Properties

keyCode Property (R)

Type ***int***
Description Code of key.

Value	Meaning
JXFS_PIN_FK_NONE	If result of a *secureReadPIN()* operation and key is not a function key.
	Otherwise:
JXFS_PIN_FK_FDK01	
JXFS_PIN_FK_FDK02	
JXFS_PIN_FK_FDK03	
JXFS_PIN_FK_FDK04	
JXFS_PIN_FK_FDK05	
JXFS_PIN_FK_FDK06	
JXFS_PIN_FK_FDK07	
JXFS_PIN_FK_FDK08	
JXFS_PIN_FK_FDK09	
JXFS_PIN_FK_FDK10	
JXFS_PIN_FK_FDK11	
JXFS_PIN_FK_FDK12	
JXFS_PIN_FK_FDK13	
JXFS_PIN_FK_FDK14	
JXFS_PIN_FK_FDK15	
JXFS_PIN_FK_FDK16	
JXFS_PIN_FK_FDK17	
JXFS_PIN_FK_FDK18	
JXFS_PIN_FK_FDK19	
JXFS_PIN_FK_FDK20	
JXFS_PIN_FK_FDK21	
JXFS_PIN_FK_FDK22	
JXFS_PIN_FK_FDK23	
JXFS_PIN_FK_FDK24	
JXFS_PIN_FK_FDK25	
JXFS_PIN_FK_FDK26	
JXFS_PIN_FK_FDK27	
JXFS_PIN_FK_FDK28	
JXFS_PIN_FK_FDK29	
JXFS_PIN_FK_FDK30	
JXFS_PIN_FK_FDK31	
JXFS_PIN_FK_FDK32	

JXFS_PIN_FK_0
JXFS_PIN_FK_1
JXFS_PIN_FK_2
JXFS_PIN_FK_3
JXFS_PIN_FK_4
JXFS_PIN_FK_5
JXFS_PIN_FK_6
JXFS_PIN_FK_7
JXFS_PIN_FK_8
JXFS_PIN_FK_9
JXFS_PIN_FK_ENTER
JXFS_PIN_FK_CANCEL
JXFS_PIN_FK_CLEAR
JXFS_PIN_FK_BACKSPACE
JXFS_PIN_FK_HELP
JXFS_PIN_FK_DECPOINT
JXFS_PIN_FK_00
JXFS_PIN_FK_000

keyType Property (R)

Type ***int***
Description Type of key pressed

It can be one of the following values:

Value	Meaning
JXFS_PIN_KP_FUNCTION	Function key.
JXFS_PIN_KP_FDKEY	Function descriptor key (FDKey).

5.6.2 Methods

JxfsPINPressedKey Constructor

Syntax ***JxfsPINPressedKey (int keyCode, int keyType)***
Description Constructor of the class.
Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	Any of the following conditions is met: *keyCode* is not one of the listed values. *keyType* is not one of the listed values.

5.7 JxfsPINReadData

This class contains the data returned by an *OperationCompleteEvent* event for *readData*() and *secureReadPIN()* operations.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
endReason	int	R	
pinLength	int	R	
pressedKeys	java.util.Vector	R	
readData	java.lang.String	R	
terminationKey	int	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINReadData	(constructor of the class)	

5.7.1 Properties

endReason Property (R)

Type ***int***
Description Indicates the input operation termination reason.

It can be one of the following values:

Value	Meaning
JXFS_PIN_COMP_AUTO	Input operation terminated because *maxLength* was reached.
JXFS_PIN_COMP_FK	A termination key was pressed.
JXFS_PIN_COMP_FDKEY	A termination FDKey was pressed

pinLength Property (R)

Type ***int***
Description If *inputMode* property is set to JXFS_PIN_INPUT_RAW, it contains the count of keys pressed.
If *inputMode* property is set to JXFS_PIN_INPUT_COOKED, it contains the count of digits entered.

pressedKeys Property (R)

Type ***java.util.Vector***
Description Vector of **JxfsPINPressedKey** objects. It represents the list of all the keys pressed during the input operation.

If *inputMode* Property (W)as set to JXFS_PIN_INPUT_RAW this property is optional and can be set to null.

readData Property (R)

Type ***java.lang.String***

Description Cooked data entered in input operation.

Value	Meaning
null	• if result of a *secureReadPIN()* operation. • if result of a *readData()* operation and *inputMode* Property was set to JXFS_PIN_INPUT_RAW.
Non formatted string representation of numeric value entered. Function keys are omitted.	• if result of a *readData()* operation and *inputMode* Property was set to JXFS_PIN_INPUT_COOKED.

terminationKey Property (R)

Type ***Int***

Description Code of termination function key or FDKey if end reason was JXFS_PIN_COMP_FK or JXFS_PIN_COMP_FDKEY.

If termination reason was JXFS_PIN_COMP_AUTO, it is set to JXFS_PIN_FK_NONE.

5.7.2 Methods

JxfsPINReadData Constructor

Syntax ***JxfsPINReadData (int endReason, int pinLength, java.util.Vector pressedKeys, java.lang.String readData, int terminationKey)***

Description Constructor of the class.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	Any of the following conditions is met: *endReason* is not one of the listed values. *pinLength* is negative. *pressedKeys* is null and inputMode is JXFS_PIN_INPUT_COOKED. *readData* is null and inputMode is JXFS_PIN_INPUT_COOKED. *terminationKey* has an invalid value.

5.8 JxfsPINFormats

This class provides properties and methods to query which PIN formats are supported by a PIN device service.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
fmt3624	boolean	R	
fmtANSI	boolean	R	
fmtISO0	boolean	R	
fmtISO1	boolean	R	
fmtEC12	boolean	R	
fmtEC13	boolean	R	
fmtEC13_Rand	boolean	R	
fmtVISA	boolean	R	
fmtDiebold	boolean	R	
fmtDieboldC0	boolean	R	

Method	Return	May use after
is*Property*	*Property*	
JxfsPINFormats	(constructor of the class)	

5.8.1 Properties

fmt3624 Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: PIN left justified, filled with padding characters, PIN length 4-16 digits.

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtANSI Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: 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).

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtISO0 Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: 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, no minimum length specified, missing digits are

filled with 0x00).

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtISO1 Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: PIN is preceded by 0x01 and the length of the PIN (0x04 to 0x0C), padding characters are taken from a transaction field (10 digits).

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtEC12 Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: similar to fmt3624, PIN only 4 digits.

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtEC13 Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: PIN is preceded by the length (digit), PIN length 4-6 digits, padded with 0x00.

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtEC13_Rand Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: PIN is preceded by the length (digit), PIN length 4-6 digits, padded with random data.

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtVISA Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports the format: same as fmtEC13.

Value	Meaning
FALSE	Format is not supported.
TRUE	Format is supported.

fmtDiebold (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports the format: PIN is padded with the padding character and may be not encrypted, single encrypted or double encrypted.

Value	**Meaning**
FALSE	Format is not supported.
TRUE	Format is supported.

fmtDieboldC0 (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports the format: PIN is preceeded by the two-digit coordination number, padded with the padding character and may be not encrypted, single encrypted or double encrypted.

Value	**Meaning**
FALSE	Format is not supported.
TRUE	Format is supported.

5.8.2 Methods

JxfsPINFormats Constructor

Syntax	***JxfsPINFormats (boolean fmt3624, boolean fmtANSI, boolean fmtSO0, boolean fmtSO1, boolean fmtEC12, boolean fmtEC13, boolean fmtEC13_Rand, boolean fmtVISA, boolean fmtDiebold, boolean fmtDieboldC0)***
Description	Constructor of the class.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PARAMETER_INVALID	All the parameters are false.

5.9 JxfsPINValidationAlgorithms

This class provides properties and methods to query which algorithms for PIN validation are supported by a PIN device service.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
valDES	boolean	R	
valEC	boolean	R	
valVISA	boolean	R	
valDESOffset	boolean	R	

Method	Return	May use after
is*Property*	*Property*	
JxfsPINValidationAlgorithms	(constructor of the class)	

5.9.1 Properties

valDES Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports DES algorithm for PIN validation.

Value	Meaning
FALSE	Algorithm is not supported.
TRUE	Algorithm is supported.

valEC Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports EUROCHEQUE algorithm for PIN validation.

Value	Meaning
FALSE	Algorithm is not supported.
TRUE	Algorithm is supported.

valVISA Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports VISA algorithm for PIN validation.

Value	Meaning
FALSE	Algorithm is not supported.
TRUE	Algorithm is supported.

valDESOffset Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports DES offset generation algorithm.

Value	Meaning

FALSE	Offset generation is not supported.
TRUE	Offset generation is supported.

5.9.2 Methods

JxfsPINValidationAlgorithms Constructor

Syntax	***JxfsPINValidationAlgorithms (boolean valDES, boolean valEC, boolean valVISA, boolean valDESOffset)***
Description	Constructor of the class.

5.10 JxfsPINChipPresentationModes

This class provides properties and methods to query which presentation algorithms for PIN chip validation are supported by a PIN device service.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
presentClear	boolean	R	

Method	Return	May use after
is*Property*	*Property*	
JxfsPINChipPresentationModes	(constructor of the class)	

5.10.1 Properties

presentClear Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports presentation of a clear text PIN to a chip card.

Value	Meaning
FALSE	Presentation algoritm is not supported.
TRUE	Presentation algoritm is supported.

5.10.2 Methods

JxfsPINChipPresentationModes Constructor

Syntax ***JxfsPINChipPresentationModes (boolean presentClear)***
Description Constructor of the class.

5.11 JxfsPINValidationData

Abstract class.

The J/XFS PIN Validation Data is the root of a hierarchy of data objects that contain data for PIN verification and used in *validatePIN(), createOffset(), createPINBlock(), validatePINSecure(), createOffsetSecure(), createPINBlockSecure()* methods of JxfsSecurePINKeypad Device Control class.

Summary

Implements : **Extends : JxfsType**

Property	Type	Access	Initialized after
validationAlgorithm	int	R	
keyName	java.lang.String	R/W	
keyEncrKey	byte[ ]	R/W	
validationTrackNumber	int	R/W	
validationLength	int	R/W	
validationIndex	int	R/W	
offsetTrackNumber	int	R/W	
offsetLength	int	R/W	
offsetIndex	int	R/W	
ejectCurrent	boolean	R/W	
ejectWhenComplete	boolean	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	

5.11.1 Properties

validationAlgorithm Property (R)

Type ***int***

Description Validation algorithm for which this object is intended to be used. Set by the constructor of each of the specific subclasses of JxfsPINValidationData to one of the following values:

Value	Meaning
JXFS_PIN_VAL_DES	DES PIN validation.
JXFS_PIN_VAL_EC	EUROCHEQUE PIN validation.
JXFS_PIN_VAL_VISA	VISA PIN validation.

keyName Property (R/W)

Type ***java.lang.String***

Description Name of the key to be used by the algorithms.
If *keyEncrKey* property is other than **null**, then this key is used to decrypt the keyEncrKey encrypted key and its results is used instead.
If *keyEncrKey* property is **null**, then this key is directly used.

For *JxfsPinBlockData* subclass, it specifies the name of the key used to encrypt the formatted PIN for the first time, or **null** if no encryption is required..

keyEncrKey Property (R/W)

Type ***byte[ ]***

Description Optional encrypted (under *keyName*) key to be used for PIN validation.

For *JxfsPinBlockData* subclass, it specifies the name of the key used to format the once encrypted formatted PIN, or **null** if no second encryption is required.

validationTrackNumber Property (R/W)

Type ***int***
Description Track where validation data is located.
Optional property.

validationLength Property (R/W)

Type ***int***
Description Length of validation data.
Optional property.

validationIndex Property (R/W)

Type ***int***
Description Location of validation data from index zero.
Optional property.

offsetTrackNumber Property (R/W)

Type ***int***
Description Track where offset data is located.
Optional property.

offsetLength Property (R/W)

Type ***int***
Description Length of offset data.
Optional property.

offsetIndex Property (R/W)

Type ***int***
Description Location of offset data from index zero.
Optional property.

ejectCurrent Property (R/W)

Type ***boolean***
Description Set true to eject any card currently in reader.
Optional property.

ejectWhenComplete Property (R/W)

Type	***boolean***
Description	Set true to eject card on completion. Optional property.

5.11.2 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for an int property is negative.

5.12 JxfsPINValidationDataForDES

Class that contains data required for DES PIN validation.

Summary

Implements : **Extends :** *JxfsPINValidationData*

Property	Type	Access	Initialized after
decimalTable	byte[ ]	R/W	
maxPIN	int	R/W	
noLeadingZero	boolean	R/W	
offset	byte[ ]	R/W	
offsetUsed	boolean	R/W	
paddingChar	byte	R/W	
validationData	byte [ ]	R/W	
validationDigits	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINValidationDataFor DES	(constructor of the class)	

5.12.1 Properties

decimalTable Property (R/W)

Type ***byte[ ]***
Description ASCII decimalization table (16 character string containing ‘0’ to ‘9’). Used to convert the hexadecimal digits (0x0 to 0xF) of the encrypted validation data to decimal digits (0x0 to 0x9).

maxPIN Property (R/W)

Type ***int***
Description Maximum number of PIN digits to be used for validation.

noLeadingZero Property (R/W)

Type ***boolean***
Description If set to TRUE and the first digit of result of the modulo 10 addition is a X’0’, it is replaced with X’1’ before performing the verification against the entered PIN. If set to FALSE, a leading zero is allowed in entered PINs.

offset Property (R/W)

Type ***byte [ ]***
Description Offset for the PIN block.
If this property is set to **null**, the offset is to be read from the card in the device.
Optional property.

offsetUsed Property (R/W)

Type	***boolean***
Description	Specifies if offset is used for PIN validation.

paddingChar Property (R/W)

Type	***byte***
Description	Specifies the padding character for validation data.

validationData Property (R/W)

Type	***byte [ ]***
Description	Validation data. If this property is set to **null**, the validation data is to be read from the card in the device.

validationDigits Property (R/W)

Type	***int***
Description	Number of Validation digits to be used for validation.

5.12.2 Methods

JxfsPINValidationDataForDES Constructor

Syntax	***JxfsPINValidationDataForDES (java.lang.String keyName, byte[ ] keyEncrKey, byte[ ] decimalTable, int maxPIN, boolean noLeadingZero, byte[ ] offset, boolean offsetUsed, byte paddingChar, byte[ ] validationData, int validationDigits)*** ***JxfsPINValidationDataForDES (java.lang.String keyName, byte[ ] keyEncrKey, int validationTrackNumber, int validationLength, int validationIndex, int offsetTrackNumber, int offsetLength, int offsetIndex, boolean ejectCurrent, ejectWhenComplete, byte[ ] decimalTable, int maxPIN, boolean noLeadingZero, byte paddingChar, byte[ ] validationData, int validationDigits)***
Description	Constructors of the class.

5.12.3 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for an int property is negative.
- The value for decimalTable is null.

5.13 JxfsPINValidationDataForEC

Class that contains data required for EUROCHEQUE PIN validation.

Summary

Implements : **Extends :** *JxfsPINValidationData*

Property	Type	Access	Initialized after
decimalTable	byte[ ]	R/W	
eurochequeData	byte[ ]	R/W	
firstEncDigits	int	R/W	
firstEncOffset	int	R/W	
PINVV	byte [ ]	R/W	
PINVVDigits	int	R/W	
PINVVOffset	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINValidationDataForEC	(constructor of the class)	

5.13.1 Properties

decimalTable Property (R/W)

Type ***byte[ ]***
Description ASCII decimalization table (16 character string containing ‘0’ to ‘9’). Used to convert the hexadecimal digits (0x0 to 0xF) of the encrypted validation data to decimal digits (0x0 to 0x9).

eurochequeData Property (R/W)

Type ***byte[ ]***
Description Track 3 Eurocheque data.

firstEncDigits Property (R/W)

Type ***Int***
Description Number of digits to extract after first encription.

firstEncOffset Property (R/W)

Type ***Int***
Description Offset of digits to extract after first encription.

PINVV Property (R/W)

Type ***byte [ ]***
Description PIN Validation Value from track data.

PINVVDigits Property (R/W)

Type	***Int***
Description	Number of validation digits to extract for PVV.

PINVVOffset Property (R/W)

Type	***Int***
Description	Offset of digits to extract for PVV.

5.13.2 Methods

JxfsPINValidationDataForEC Constructor

Syntax	***JxfsPINValidationDataForEC (java.lang.String keyName, byte[ ] keyEncrKey, byte[ ] decimalTable, byte[ ] eurochequeData, int firstEncDigits, int firstEncOffset, byte[ ] PINVV, int PINVVDigits, int PINVVOffset)***
Description	Constructor of the class.

5.13.3 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for an int property is negative.
- The value for decimalTable, eurochequeData or PINVV is null.

5.14 JxfsPINValidationDataForVISA

Class that contains data required for VISA PIN validation.

Summary

Implements : **Extends :** *JxfsPINValidationData*

Property	Type	Access	Initialized after
PAN	byte[ ]	R/W	
PINVV	byte[ ]	R/W	
PINVVDigits	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINValidationDataForVISA	(constructor of the class)	

5.14.1 Properties

PAN Property (R/W)

Type ***byte[ ]***
Description Primary Account Number from track data.

PINVV Property (R/W)

Type ***byte[ ]***
Description PIN Validation Value from track data.

PINVVDigits Property (R/W)

Type ***int***
Description Number of digits of PVV.

5.14.2 Methods

JxfsPINValidationDataForVISA Constructor

Syntax ***JxfsPINValidationDataForVISA (java.lang.String keyName, byte[ ] keyEncrKey, byte[ ] PAN, byte[ ] PINVV, byte[ ] PINVVDigits)***
Description Constructor of the class.

5.14.3 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for PINVVDigits is negative or zero.
- The value for PAN or PINVV is null.

5.15 JxfsPINOffsetData

Data class for data required for createOffset() method of JxfsSecurePINKeypad.

Summary

Implements : **Extends :** *JxfsPINValidationData*

Property	Type	Access	Initialized after
decimalTable	byte[ ]	R/W	
maxPIN	int	R/W	
paddingChar	byte	R/W	
validationData	byte[ ]	R/W	
validationDigits	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINOffsetData	(constructor of the class)	

5.15.1 Properties

decimalTable Property (R/W)

Type ***byte[ ]***
Description ASCII decimalization table (16 position byte array containing '0' to '9' characters). Used to convert the hexadecimal digits (0x0 to 0xF) of the encrypted validation data to decimal digits (0x0 to 0x9).

maxPIN Property (R/W)

Type ***int***
Description Maximum number of PIN digits to be used for validation.

paddingChar Property (R/W)

Type ***byte***
Description Specifies the padding character for validation data.

validationData Property (R/W)

Type ***byte[ ]***
Description Validation data.
If this property is set to **null**, the validation data is to be read from the card in the device.

validationDigits Property (R/W)

Type ***int***
Description Number of Validation digits to be used for validation.

5.15.2 Methods

JxfsPINOffsetData Constructor

Syntax	***JxfsPINOffsetData (java.lang.String keyName, byte[ ] keyEncrKey, byte[ ] decimalTable, int maxPIN, byte paddingChar, byte[ ] validationData, int validationDigits)*** ***JxfsPINOffsetData (java.lang.String keyName, byte[ ] keyEncrKey, int validationTrackNumber, int validationLength, int validationIndex, boolean ejectCurrent, ejectWhenComplete, byte[ ] decimalTable, int maxPIN, byte paddingChar, int validationDigits)***
Description	Constructor of the class.

5.15.3 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for maxPIN or validationDigits is negative or zero.
- The value for decimalTable is null.

5.16 JxfsPINBlockData

Data class for data required for *pinBlock()* method of JxfsSecurePINKeypad.

Summary

Implements : **Extends :** *JxfsPINValidationData*

Property	Type	Access	Initialized after
customerData	byte[ ]	R/W	
paddingChar	byte	R/W	
pinBlockFormat	int	R/W	
XORData	byte[ ]	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINBlockData	(constructor of the class)	

5.16.1 Properties

customerData Property (R/W)

Type ***byte[ ]***
Description Used for ANSI, ISO-0 and ISO-1 algorithm to build the formatted PIN. For ANSI and ISO-0 the PAN (Primary Account Number) is used, for ISO-1 a ten digit transaction field is required. If not used a **null** is required.
Used for DIEBOLD with coordination number, as a two digit coordination number.
If this property is set to **null**, the validation data is to be read from the card in the device.

paddingChar Property (R/W)

Type ***byte***
Description Specifies the padding character.

pinBlockFormat Property (R/W)

Type ***int***
Description Specifies the format of the PIN block.
Possible values are:

Value	Meaning
JXFS_PIN_FMT_3624	Format 3624.
JXFS_PIN_FMT_ANSI	Format ANSI.
JXFS_PIN_FMT_ISO0	Format ISO0.
JXFS_PIN_FMT_ISO1	Format ISO1.
JXFS_PIN_FMT_EC12	Format EC12.
JXFS_PIN_FMT_EC13	Format EC13.
JXFS_PIN_FMT_EC13RAND	Format EC13, random padding.
JXFS_PIN_FMT_VISA	Format VISA.
JXFS_PIN_FMT_DIEBOLD	Format DIEBOLD.
JXFS_PIN_FMT_DIEBOLDC0	Format DIEBOLD C0.

XORData Property (R/W)

Type	***byte[ ]***
Description	If the formatted PIN is encrypted twice to build the resulting PIN block, this data can be used to modify the result of the first encryption by an XOR-operation..

5.16.2 Methods

JxfsPINBlockData Constructor

Syntax	***JxfsPINBlocktData (java.lang.String keyName, byte[ ] keyEncrKey, byte[ ] customerData, byte paddingChar, int pinBlockFormat, byte[ ] XORData)*** ***JxfsPINBlocktData (java.lang.String keyName, byte[ ] keyEncrKey, int validationTrackNumber, int validationLength, int validationIndex,, boolean ejectCurrent, ejectWhenComplete, byte paddingChar, int pinBlockFormat, byte[ ] XORData)***
Description	Constructor of the class.

5.16.3 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for pinBlockFormat is out of range.
- The value for XORData is null.

5.17 JxfsPINChipValidationData

Abstract class.

The J/XFS PIN Chip Validation Data is the root of a hierarchy of data objects that contain data for PIN chip verification and used in *validationPINChip()* method of JxfsSecurePINKeypad Device Control class.

Summary

Implements : **Extends : JxfsType**

Property	Type	Access	Initialized after
presentationMode	int	R/W	
chipProtocol	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	

5.17.1 Properties

presentationMode Property (R/W)

Type ***int***
Description Presentation mode for which this object is intended to be used.

Set by the constructor of each of the specific subclasses of *JxfsPINChipValidationData*.

Possible values are:

Value	Meaning
JXFS_PIN_PRES_CLEAR	Clear text presentation of PIN to chip card device.

chipProtocol Property (R/W)

Type ***int***
Description Protocol to be used with chip.
Possible values are:

Value	Meaning
0 .. 15	Protocols T=0 .. T=15.

5.17.2 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for presentationMode or chipProtocol is out of range.

5.18 JxfsPINChipValidationDataClear

Class that contains data required for Clear chip PIN validation.

Summary

Implements :

Extends :
JxfsPINChipValidationData

Property	Type	Access	Initialized after
chipData	byte[ ]	R/W	
insertPosition	int	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINChipValidationDataClear	(constructor of the class)	

5.18.1 Properties

chipData Property (R/W)

Type ***byte[ ]***
Description Data to be sent to the chip.

insertPosition Property (R/W)

Type ***int***
Description Contains the bit position where to insert the PIN in the *chipData* buffer (0 means is bit 0 of first byte, and so on).

5.18.2 Methods

JxfsPINChipValidationDataClear Constructor

Syntax ***JxfsPINChipValidationDataClear (int chipProtocol, byte[ ] chipData, int insertPosition)***
Description Constructor of the class.

5.18.3 Exceptions

Exception JXFS_E_PARAMETER_INVALID is thrown by the setter methods in the following cases:

- The value for insertPosition is negative.
- The value for chipData is null.

5.19 JxfsPINValidationResult

This class contains the result of a PIN validation operation.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
validationResult	boolean	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINValidationResult	(constructor of the class)	

5.19.1 Properties

validationResult Property (R)

Type ***boolean***
Description TRUE if PIN was validated, otherwise FALSE.

5.19.2 Methods

JxfsPINValidationResult Constructor

Syntax ***JxfsPINValidationResult (boolean validationResult)***
Description Constructor of the class.

5.20 JxfsPINOffset

This class contains a PIN offset.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
offsetValue	byte[ ]	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINOffset	(constructor of the class)	

5.20.1 Properties

offsetValue Property (R)

Type ***byte[ ]***
Description A PIN Offset

5.20.2 Methods

JxfsPINOffset Constructor

Syntax ***JxfsPINOffset (byte[ ] offsetValue)***
Description Constructor of the class.
Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	*offsetValue* is null.

5.21 JxfsPINBlock

This class contains a PIN block.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
PINBlockValue	byte[ ]	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINBlock	(constructor of the class)	

5.21.1 Properties

PINBlockValue Property (R)

Type ***byte[ ]***
Description A PIN Block.

5.21.2 Methods

JxfsPINBlock Constructor

Syntax ***JxfsPINBlock (byte[ ] PINBlockValue)***
Description Constructor of the class.

5.22 JxfsPINChipValidationResult

This class contains the result of a PIN chip validation operation.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
validationResult	byte[ ]	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINChipValidationResult	(constructor of the class)	

5.22.1 Properties

validationResult Property (R)

Type ***byte[ ]***
Description Data returned from chip.

5.22.2 Methods

JxfsPINChipValidationResult Constructor

Syntax ***JxfsPINChipValidationResult (byte[ ] validationResult)***
Description Constructor of the class.
Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	*validationResult* is null.

5.23 JxfsPINCryptoModes

This class provides properties and methods to query which encyption modes are supported by a secure PIN device service.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
cryptDESECB	boolean	R	
cryptDESCBC	boolean	R	
cryptDESCFB	boolean	R	
cryptDESMAC	boolean	R	
cryptRSA	boolean	R	
cryptECMA	boolean	R	
cryptTRIDESECB	boolean	R	
cryptTRIDESCBC	boolean	R	
cryptTRIDESCFB	boolean	R	
cryptTRIDESMAC	boolean	R	

Method	Return	May use after
is*Property*	*Property*	
JxfsPINCryptoModes	(constructor of the class)	

5.23.1 Properties

cryptDESECB Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports Electronic Code Book encryption.

Value	Meaning
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptDESCBC Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports Cypher Block Chaining encryption.

Value	Meaning
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptDESCFB Property (R)

Type ***boolean***
Initial Value Depends on device
Description Indicates if the device supports Cypher Feed Back encryption.

Value	Meaning
FALSE	Encryption mode is not supported.

TRUE	Encryption mode is supported.

cryptDESMAC Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports MAC calculation using CBC.

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptRSA Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports RSA encryption.

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptECMA Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports ECMA encryption.

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptTRIDESECB Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports Triple DES with Electronic Code Book.

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptTRIDESCBC Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports Triple DES with Cypher Block Chaining.

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptTRIDESCFB Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports Triple DES with Cypher Feed Back.

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

cryptTRIDESMAC Property (R)

Type	***boolean***
Initial Value	Depends on device
Description	Indicates if the device supports Triple DES MAC calculation using CBC .

Value	**Meaning**
FALSE	Encryption mode is not supported.
TRUE	Encryption mode is supported.

5.23.2 Methods

JxfsPINCryptoModes Constructor

Syntax	***JxfsPINCryptoModes (boolean cryptDESECB, boolean cryptDESCBC, boolean cryptDESCFB, boolean cryptDESMAC, boolean cryptRSA, boolean cryptECMA, boolean cryptTRIDESECB, boolean cryptTRIDESCBC, boolean cryptTRIDESCFB, boolean cryptTRIDESMAC)***
Description	Constructor of the class.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PARAMETER_INVALID	All the parameters are false.

5.24 JxfsPINKeyDetail

The J/XFS PIN Key Detail data class contains relevant information for an application about a key in the device's key table.

Summary

Implements : **Extends : JxfsType**

Property	Type	Access	Initialized after
keyLoaded	boolean	R	
keyName	java.lang.String	R	
keyReload	boolean	R	
keyUse	**JxfsPINKeyUses**	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINKeyDetail	(constructor of the class)	

5.24.1 Properties

keyLoaded Property (R)

Type ***boolean***
Description Indicates wether the key has been loaded/imported.

Value	Meaning
TRUE	Key has been loaded/imported and is ready to be used.
FALSE	Key is not operationally ready.

keyName Property (R)

Type ***jave.lang.String***
Description Name of the key.

keyReload Property (R)

Type ***boolean***
Description Indicates wether the key can be loaded/imported just once.

Value	Meaning
TRUE	Key can be loaded/imported.
FALSE	Key can only be loaded/imported once.

keyUse Property (R)

Type ***JxfsPINKeyUses***
Description Type of access for which the key is intended to be used.

5.24.2 Methods

JxfsPINKeyDetail Constructor

Syntax ***JxfsPINKeyDetail (boolean keyLoaded, java.lang.String keyName, boolean keyReload, JxfsKeyUses keyUse)***

Description Constructor of the class.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	Any of the following conditions is met: *keyName* is null. *keyUse* is null.

5.25 JxfsPINKeyToImport

The J/XFS PIN Key to Import data class contains data required as input for *importKey()* operation.

Summary

Implements : **Extends : JxfsType**

Property	Type	Access	Initialized after
key	java.lang.String	R/W	
keyEncKey	java.lang.String	R/W	
keyReload	boolean	R/W	
keyUse	**JxfsPINKeyUses**	R/W	
keyValue	byte[ ]	R/W	
idKey	byte[ ]	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINKeyToImport	(constructor of the class)	

5.25.1 Properties

key Property (R/W)

Type ***java.lang.String***
Description Name of the key being loaded.

keyEncKey Property (R/W)

Type ***java.lang.String***
Description Name of the key encrypting key that was used to encrypt the *keyValue* property data.
If this property is set to null, the key specified in *keyValue* is directely stored in the device‘s key table.

keyReload Property (R/W)

Type ***boolean***
Description Indicates wether the key can be loaded only once.

Value	Meaning
TRUE	Key can be loaded/imported may times.
FALSE	Key can only be loaded/imported once.

keyUse Property (R/W)

Type ***JxfsPINKeyUses***
Description Type of access for which the key is intended to be used.

keyValue Property (R/W)

Type ***byte[ ]***
Description Key value.

idKey Property (R/W)

Type ***byte[ ]***

Description Specifies the key owner identification or null.

5.25.2 Methods

JxfsPINKeyToImport Constructor

Syntax ***JxfsPINKeyToImport (java.lang.String key, java.lang.String keyEncKey, boolean keyReload, JxfsKeyUses keyUse, byte[ ] keyValue, byte[ ] idKey)***

Description Constructor of the class.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	Any of the following conditions is met: *key* is null. *keyUse* is null. *keyValue* is null. *idKey* is null.

5.26 JxfsPINInitialization

This class contains the result of a security module's initialization operation.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
idKey	byte[ ]	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINInitialization	(constructor of the class)	

5.26.1 Properties

idKey Property (R)

Type ***byte[ ]***
Description Value of the ID key encrypted by the ID encryption key. Can be used as authorization for importKey() method.

Null if not supported by the device.

5.26.2 Methods

JxfsPINInitialization Constructor

Syntax ***JxfsPINInitialization (byte[ ] idKey)***
Description Constructor of the class.

5.27 JxfsPINKeyVerificationData

This class contains data returned after the completion of a *importKey()* operation..

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
keyVerCode	byte[ ]	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINKeyVerificationData	(constructor of the class)	

5.27.1 Properties

keyVerCode Property (R)

Type ***byte[ ]***
Description Key verification code data that can be used for verification of the loaded key.
Null if this function is not supported by the device.

5.27.2 Methods

JxfsPINKeyVerificationData Constructor

Syntax ***JxfsPINKeyVerificationData (byte[ ] keyVerCode)***
Description Constructor of the class.

5.28 JxfsPINCryptoData

The J/XFS PIN Cryptographic data class contains data required for encryption/decryption methods.

Summary

Implements : **Extends : JxfsType**

Property	Type	Access	Initialized after
cryptoMode	int	R/W	
data	byte[ ]	R/W	
key	java.lang.String	R/W	
keyEncKey	java.lang.String	R/W	
paddingChar	byte	R/W	
startValue	byte[ ]	R/W	
startValueKey	java.lang.String	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINCryptoData	(constructor of the class)	

5.28.1 Properties

cryptoMode Property (R/W)

Type ***int***

Description Indicates the algorithm to be used.

Value	Meaning
JXFS_PIN_CRYPT_MODE_DESECB	Electronic Code Book
JXFS_PIN_CRYPT_MODE_DESCBC	Cipher Block Chaining
JXFS_PIN_CRYPT_MODE_DESMAC	MAC calculation using CBC
JXFS_PIN_CRYPT_MODE_DESCFB	Cipher Feed Back
JXFS_PIN_CRYPT_MODE_RSA	RSA Encryption
JXFS_PIN_CRYPT_MODE_ECMA	ECMA Encryption
JXFS_PIN_CRYPT_MODE_TRIDESECB	Triple DES with Electronic Code Book
JXFS_PIN_CRYPT_MODE_TRIDESCBC	Triple DES with Cipher Block Chaning
JXFS_PIN_CRYPT_MODE_TRIDESCFB	Triple DES with Cipher Feed Back
JXFS_PIN_CRYPT_MODE_TRIDESMAC	Triple DES MAC calculation using CBC

data Property (R/W)

Type ***byte[ ]***

Description Data to be encrypted, decrypted or MACed.

key Property (R/W)

Type	***java.lang.String***
Description	Name of the key to be used in cryptographic operation.

keyEncKey Property (R/W)

Type	***java.lang.String***
Description	Encrypted key, under the key contained in *key* property, to be used in cryptographic operation. If null, key contained in *key* property is used.

paddingChar Property (R/W)

Type	***byte***
Description	Specifies the padding character used.

startValue Property (R/W)

Type	***byte[ ]***
Description	DES and Triple DES initialization vector for the CBC, CFB and MAC. If null, s*tartValueKey* property is used as the Initialization Vector. If both are null the default is 16 hexadecimal digits 0x00.

startValueKey Property (R/W)

Type	***java.lang.String***
Description	Name of the stored key used to decrypt the s*tartValue* property to obtain the Initialization Vector. If null, *startValue* is used as the initialization vector.

5.28.2 Methods

JxfsPINCryptoData Constructor

Syntax	***JxfsPINCryptoData (int cryptoMode, byte[ ] data, java.lang.String key, java.lang.String keyEncKey, byte paddingChar, byte[ ] startValue, java.lang.String startValueKey)***
Description	Constructor of the class.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	**Meaning**
JXFS_E_PARAMETER_INVALID	Any of the following conditions is met: *cryptoMode* is out of range. *data* is null. *key* is null.

5.29 JxfsPINMACData

The J/XFS PIN Cryptographic MAC data class contains data required for MAC generation operation.
It is a subclass of *JxfsPINCryptoData*.

Summary

Implements : **Extends : JxfsPINCryptoData**

Property	Type	Access	Initialized after
compression	boolean	R/W	
compressionChar	byte	R/W	

Method	Return	May use after
get*Property*	*Property*	
set*Property*	*void*	
JxfsPINMACData	(constructor of the class)	

5.29.1 Properties

compression Property (R/W)

Type ***boolean***
Description Specifies wether data is to be compressed (blanks removed) before building the MAC.

compressionChar Property (R/W)

Type ***byte***
Description If compression is **TRUE**, it specifies the representation of the blank character in the actual code table.

5.29.2 Methods

JxfsPINMACData Constructor

Syntax ***JxfsPINMACData (boolean compression, byte compressionChar)***
Description Constructor of the class.

5.30 JxfsPINCryptoResult

The J/XFS PIN Cryptographic result data class contains data data returned by cryptographic operations (encrypt, decrypt and generateMAC).

Summary

Implements : **Extends : JxfsType**

Property	Type	Access	Initialized after
cryptoResult	byte[ ]	R	

Method	Return	May use after
get*Property*	*Property*	
JxfsPINCryptoResult	(constructor of the class)	

5.30.1 Properties

cryptoResult Property (R/W)

Type ***byte[ ]***
Description Data returned by a cryptographic operation.

5.30.2 Methods

JxfsPINCryptoResult Constructor

Syntax ***JxfsPINCryptoResult (byte[ ] cryptoResult)***
Description Constructor of the class.
Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	*cryptoResult* is null.

5.31 JxfsPINKeyUses

Summary

This class provides properties and methods to query which type of access a key is intended for.

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
kuseEncDec	boolean	R/W	
kusePin	boolean	R/W	
kuseMac	boolean	R/W	
kuseKek	boolean	R/W	
kuseVek	boolean	R/W	
kuseMaster	boolean	R/W	

Method	Return	May use after
is*Property*	*Property*	
set*Property*	*void*	
JxfsPINKeyUses	(constructor of the class)	

5.31.1 Properties

kuseEncDec Property (R/W)

Type ***boolean***
Description Indicates if the key may be used for encryption and decryption.

Value	Meaning
FALSE	This use is not supported.
TRUE	This use is supported.

kusePin Property (R/W)

Type ***boolean***
Description Indicates if the key may be used for PIN functions.

Value	Meaning
FALSE	This use is not supported.
TRUE	This use is supported.

kuseMac Property (R/W)

Type ***boolean***
Description Indicates if the key may be used for MAC generation.

Value	Meaning
FALSE	This use is not supported.
TRUE	This use is supported.

kuseKek Property (R/W)

Type ***boolean***
Description Indicates if the key may be used as key encryption key.

Value	Meaning
FALSE	This use is not supported.
TRUE	This use is supported.

kuseVek Property (R/W)

Type ***boolean***

Description Indicates if the key may be used as CBC Start Value encryption key.

Value	Meaning
FALSE	This use is not supported.
TRUE	This use is supported.

kuseMaster Property (R/W)

Type ***boolean***

Description Indicates if the key may be used as Master encryption key.

Value	Meaning
FALSE	This use is not supported.
TRUE	This use is supported.

5.31.2 Methods

JxfsPINKeyUses Constructor

Syntax ***JxfsPINKeyUses (boolean kuseEncDec, boolean kusePin, boolean kuseMac, boolean kuseKek, boolean kuseVek, boolean kuseMaster)***

Description Constructor of the class.

Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	All the parameters are false.

5.32 JxfsPINIdKeyModes

This class provides properties and methods to query which type of uses of ID keys are implemented.

Summary

Implements : **Extends :** **JxfsType**

Property	Type	Access	Initialized after
idKeyInitialize	boolean	R	
idKeyImport	boolean	R	

Method	Return	May use after
is*Property*	*Property*	
JxfsPINIdKeyModes	(constructor of the class)	

5.32.1 Properties

idKeyInitialize Property (R)

Type ***boolean***
Description ID key is supported in the Initialize method.

Value	Meaning
FALSE	Feature is not supported.
TRUE	Feature is supported.

idKeyImport Property (R)

Type ***boolean***
Description ID key is supported in the ImportKey method.

Value	Meaning
FALSE	Feature is not supported.
TRUE	Feature is supported.

5.32.2 Methods

JxfsPINIdKeyModes Constructor

Syntax ***JxfsPINIdKeyModes (boolean idKeyInitialize, boolean idKeyImport)***
Description Constructor of the class.
Exceptions Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes.

Value	Meaning
JXFS_E_PARAMETER_INVALID	All the parameters are false.

6 Codes

6.1 Error Codes

Value	Meaning
JXFS_E_PIN_READ_FAILURE	Read error.
JXFS_E_PIN_KEYINVALID	At least one of the specified active function keys or FDKeys is invalid.
JXFS_E_PIN_NOACTIVEKEYS	No active function key or FDKey specified.
JXFS_E_PIN_KEYNOTSUPPORTED	At least one of the specified active function keys or FDKeys (*activeFKeys* or *activeFDKeys* properties of *readMode* parameter) is not supported by the device service.
JXFS_E_PIN_MINIMUNLENGTH	The *minLength* property is invalid or greater than the *maxLength* property.
JXFS_E_PIN_NO_PIN	PIN has not been entered or has been cleared.
JXFS_E_PIN NOT_ALLOWED	PIN entered by the user is not allowed.
JXFS_E_PIN_KEY_NOT_FOUND	The specified key was not found.
JXFS_E_PIN_KEY_NO_VALUE	The specified key is not loaded.
JXFS_E_PIN_USE_VIOLATION	The specified use is not supported by this key.
JXFS_E_PIN_ACCESS_DENIED	The encryption module is either not initialized or not ready for any vendor specific reason.
JXFS_E_PIN_NOTSUPPORTEDCAP	The requested function is not supported.
JXFS_E_PIN_FORMAT_NOTSUPPORTED	The specified PIN block format is not supported.
JXFS_E_PIN_LENGTH_ERROR	The length of the start value specified is not supported.
JXFS_E_PIN_CRYPTNOTSUPPORTED	The encryption or decryption method is not supported.
JXFS_E_PIN_DUPLICATE_KEY	A key exists with the specified name and cannot be overwritten.

6.2 Status Codes

Value	Meaning
JXFS_S_PIN_KEY	A new key has been loaded/imported into the device's key table.

6.3 Operation Codes

The following codes identify the operation that generated an OperationCompleteEvent or IntermediateEvent:

Value	Method
JXFS_O_PIN_READPIN	*readData, secureReadPIN*
JXFS_O_PIN_CREATEOFFSET	*createOffset*
JXFS_O_PIN_CREATEPINBLOCK	*createPINBlock*
JXFS_O_PIN_VALIDATEPIN	*validatePIN*
JXFS_O_PIN_CREATEOFFSET_SECURE	*createOffsetSecure*
JXFS_O_PIN_CREATEPINBLOCK_SECURE	*createPINBlockSecure*

JXFS_O_PIN_VALIDATEPIN_SECURE	*validatePINSecure*
JXFS_O_PIN_VALIDATEPINCHIP	*validatePINChip*
JXFS_O_PIN_DECRYPT	*decrypt*
JXFS_O_PIN_ENCRYPT	*encrypt*
JXFS_O_PIN_GENMAC	*generateMAC*
JXFS_O_PIN_IMPORTKEY	*importKey*
JXFS_O_PIN_INITIALIZE	*initialize*

The following codes identify the reason for an IntermediateEvent:

Value	Meaning
JXFS_I_PIN_KEY_PRESSED	A key has been pressed.

6.4 Constants

Value	Meaning
JXFS_PIN_FK_FDK01 to JXFS_PIN_FK_FDK32	Codes of function descriptor keys FDKeys.
JXFS_PIN_FK_0	Function key code.
JXFS_PIN_FK_1	Function key code.
JXFS_PIN_FK_2	Function key code.
JXFS_PIN_FK_3	Function key code.
JXFS_PIN_FK_4	Function key code.
JXFS_PIN_FK_5	Function key code.
JXFS_PIN_FK_6	Function key code.
JXFS_PIN_FK_7	Function key code.
JXFS_PIN_FK_8	Function key code.
JXFS_PIN_FK_9	Function key code.
JXFS_PIN_FK_ENTER	Function key code.
JXFS_PIN_FK_CANCEL	Function key code.
JXFS_PIN_FK_CLEAR	Function key code.
JXFS_PIN_FK_BACKSPACE	Function key code.
JXFS_PIN_FK_HELP	Function key code.
JXFS_PIN_FK_DECPOINT	Function key code.
JXFS_PIN_FK_00	Function key code.
JXFS_PIN_FK_000	Function key code.
JXFS_PIN_FK_NONE	Result of a *secureReadPIN()* operation when key is not a function key.
JXFS_PIN_KP_FUNCTION	Key is a Function key.
JXFS_PIN_KP_FDKEY	Key is a Function descriptor key (FDKey).
JXFS_PIN_INPUT_RAW	Each key pressed during an input operation will generate an intermediate event. These events will contain information about pressed keys.
JXFS_PIN_INPUT_COOKED	No intermediate events per key pressed are generated. Data entered during an input operation is provided in an *OperationCompleteEvent* event.
JXFS_PIN_COMP_AUTO	Input operation terminated because *maxLength* was reached.
JXFS_PIN_COMP_FK	A termination key was pressed.
JXFS_PIN_COMP_FDKEY	A termination FDKey was pressed

Value	Meaning
JXFS_PIN_VAL_DES	DES PIN validation.
JXFS_PIN_VAL_EC	EUROCHEQUE PIN validation.

JXFS_PIN_VAL_VISA	VISA PIN validation.
JXFS_PIN_PRES_CLEAR	Clear text presentation of PIN to chip card device.

PIN block formats:

Value	Meaning
JXFS_PIN_FMT_3624	3624.
JXFS_PIN_FMT_ANSI	ANSI.
JXFS_PIN_FMT_ISO0	ISO0.
JXFS_PIN_FMT_ISO1	ISO1.
JXFS_PIN_FMT_EC12	EC12.
JXFS_PIN_FMT_EC13	EC13.
JXFS_PIN_FMT_EC13RAND	EC13, random padding.
JXFS_PIN_FMT_VISA	VISA.
JXFS_PIN_FMT_DIEBOLD	DIEBOLD.
JXFS_PIN_FMT_DIEBOLDC0	DIEBOLD C0.

Encryption/decryption algorithms:

Value	Meaning
JXFS_PIN_CRYPT_MODE_DESECB	Electronic Code Book
JXFS_PIN_CRYPT_MODE_DESCBC	Cipher Block Chaining
JXFS_PIN_CRYPT_MODE_DESMAC	MAC calculation using CBC
JXFS_PIN_CRYPT_MODE_DESCFB	Cipher Feed Back
JXFS_PIN_CRYPT_MODE_RSA	RSA Encryption
JXFS_PIN_CRYPT_MODE_ECMA	ECMA Encryption
JXFS_PIN_CRYPT_MODE_TRIDESECB	Triple DES with Electronic Code Book
JXFS_PIN_CRYPT_MODE_TRIDESCBC	Triple DES with Cipher Block Chaning
JXFS_PIN_CRYPT_MODE_TRIDESCFB	Triple DES with Cipher Feed Back
JXFS_PIN_CRYPT_MODE_TRIDESMAC	Triple DES MAC calculation using CBC