

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

WORKSHOP AGREEMENT

CWA 13937-3

August 2000

ICS 35.240.40

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part 3: Magnetic Stripe & Chip Card Device Class Interface - Programmer's Reference

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

© 2000 CEN All rights of exploitation in any form and by any means reserved world-wide for CEN National Members

Ref. No CWA 13937-3:2000 E

Page 2 CWA 13937-3:2000

Foreword

This CWA contains the specifications that define the J/eXtensions for Financial Services (J/XFS) for the Java TM 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

Note:
 Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. The Java Trademark Guidelines are currently available on the web at http://java.sun.com/nav/business/trademark guidelines.html.

 All other trademarks are trademarks of their respective owners.

Contents

1	SC	OPE	4
2	OV	ERVIEW	5
	2.1	DESCRIPTION	5
	2.2	CLASSES AND INTERFACES	7
	2.3	SUPPORT CLASSES	8
3	DE	VICE BEHAVIOR	9
	3.1	DEVICE OPEN()	9
4	CL	ASSES AND INTERFACES	10
	4.1	ACCESS TO PROPERTIES	10
	4.2	EXCEPTIONS	
	4.3	IJXFSMAGSTRIPECONTROL	11
	4.4	IJXFSCHIPCARDCONTROL	15
	4.5	IJxfsMotorizedCard	19
	4.6	IJXFSMSDSECURE	23
5	SUI	PPORT CLASSES	27
	5.1	JxfsMSDTracks	27
	5.2	JXFSMSDTRACKSELECTION	29
	5.3	JXFSMSDREADDATA	
	5.4	JXFSCCDDATA	32
	5.5	JXFSMSDWMDATA	
	5.6	JXFSMSDSECUREMODE	34
	5.7	JXFSMSDREADDATASECURE	35
6	CO	DES	
	6.1	Error Codes	
	6.2	Status Codes	
	6.3	OPERATION CODES	
	6.4	CONSTANTS	

1 Scope

This document describes the Magnetic Stripe Device (MSD) as well as Chip Card Device (CCD) 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 Magnetic Stripe devices and Chip Card 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 document describes the J/XFS support classes for both Magnetic Stripe devices (MSD) as well as Chip Card devices (CCD).

As well as the rest of J/XFS device controls, J/XFS Magnetic Stripe and J/XFS Chip Card devices use the event driven model and the same behavioral model. Therefore, in the case of a Magnetic Stripe device, the application will instantiate a J/XFS Magnetic Stripe Device Control Object and then use the available methods to do I/O. When an I/O method is called, the J/XFS Magnetic Stripe 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 Magnetic Stripe Device Control Object for the various types of events it wishes to handle.

The same model applies to all J/XFS device controls and, in particular, to the Chip Card Device control.

2.1.1 Magnetic Stripe Device

The J/XFS Magnetic Stripe Reader/Encoder Device Support allows for the operation of devices with magnetric stripe read/write capabilities. Following are typical devices with such a capability:

- motor driven card reader/writer
- pull through card reader/writer
- dip card reader/writer

The following tracks and the corresponding international standards are taken into account in this document:

Track 1	ISO 7811
Track 2	ISO 7811
Track 3	ISO 7811 / ISO 4909

In addition to the pure reading of the tracks mentioned above, security boxes can be used via this service to check the data of writable tracks for manipulation. These boxes (such as CIM or MM) are sensor-equipped devices that are able to check some other information on the card and compare it with the track data.

Leds handling will be defined based on initialization configuration so no reference to them is made in this document.

Handling of *watermark* is also considered.

2.1.2 Chip Card Device

The J/XFS Chip Card Device Support allows for the operation of devices with chip access capabilities. Following are typical devices with such a capability :

- motor driven chip card devices.
- dip chip card devices.

The following chips and the corresponding international standards are taken into account in this document:

• Chip (contacted) ISO 7816

2.2 Classes and Interfaces

The following classes and interfaces are used by the J/XFS MSD and CCD Device Controls. In order to support the definition of the different properties of the different devices (see Introduction), the Device Controls are defined in a class hierarchy.

Class or Interface	Name	Description	Extends or Implements
Interface	IJxfsBaseControl	Base interface for all the	
		device controls. Contains	
		methods common to all the	
		device controls.	
Interface	IJxfsMagStripeControl	Base interface for MSD	Extends:
		controls. Contains method	IJxfsBaseControl
		declarations specific to MSD	
		controls.	
Interface	IJxfsChipCardControl	Base interface for CCD	Extends:
	-	controls.	IJxfsBaseControl
		Contains method declarations	
		specific to CCD controls.	
Interface	IJxfsMotorizedCard	Interface for motorized card	
		devices.	
		Contains method declarations	
		specific to motorized card	
		devices.	
Interface	IJxfsMSDSecure	Interface for motorized card	Extends:
		devices with secure module.	IJxfsMotorizedCard
		Contains method declarations	
		specific to card devices with	
		secure module.	
Class	JxfsBaseControl	Base class for all the device	
		controls. Contains properties	
		common to all the deviceb	
		controls.	
Class	JxfsMagStripe	Base class for MSD controls.	Implements:
		Contains properties specific to	IjxfsMagStripeControl
		MSD device controls.	IJxfsMSDSecure
Class	JxfsChipCard	Base class for CCD controls.	Implements:
		Contains properties specific to	IjxfsChipCardControl
		CCD device controls.	IJxfsMotorizedCard

2.3 Support Classes

Class or Inter-face	Name	Description	Extends / Implements
Interface	JxfsConst	Interface containing the Jxfs	
		constants that are common to	
		several device categories	
Interface	JxfsMSDConst	Interface containing the Jxfs	
		constants that are common to	
		all the MSD device controls.	
Interface	JxfsCCDConst	Interface containing the Jxfs	
		constants that are common to	
		all the CCD device controls.	
Interface	JxfsMotorizedCardConst	Interface containing the Jxfs	
		constants for motorized card	
		devices.	
Class	JxfsMSDTracks	MSD Track selector class.	Extends:
		Indicates for each track if its	JxfsType
		selected or not.	
		Properties are read only.	
Class	JxfsMSDTrackSelection	Subclass of MSD Track	Extends:
		selector class. It contains the	JxfsMSDTracks
		same properties but they can	
		be set by applications.	
Class	JxfsMSDReadData	Data class that contains data	Extends:
		returned in Operation	JxfsType
		Complete events for MSD	
		<i>readData()</i> operation.	
Class	JxfsCCDData	Data class that contains data	Extends:
		returned in Operation	JxfsType
		Complete events for CCD	
		input/output operations.	
Class	JxfsMSDWmData	Data class that contains data	Extends:
		returned in Operation	JxfsType
		Complete events for MSD	
		<i>readWMtrack()</i> operation.	
Class	JxfsMSDSecureMode	Data class that provides	Extends:
		required properties for	JxfsType
		<i>readData()</i> operation in	
		secure mode.	
Class	JxfsMSDReadDataSecure	Data class that contains data	Extends:
		returned in Operation	JxfsType
		Complete events for MSD	
		<i>readData()</i> in secure mode.	
Class	JxfsEvent	Abstract class from which all	Extends:
		Jxfs event classes are	java.util.
		extended	EventObject
Class	EventEvent	The Device Service creates	Extends:
		<i>Event</i> event instances of this	JxfsEvent
		class and delivers them	
		through the J/XFS MSD	
		Device Control's event	
		callbacks to the application	
Class	JxfsException	Exception class. The J/XFS	Extends:
C1000	JAISLACEPHUII	MSD Device Control creates	java.lang.Exception
		and throws exceptions on	Javanang.DACepuon
		method failure and property	
		access failure.	
		access failure.	l

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.

get Property

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:		
-	JXFS_E_CLOSED		
	JXFS_E_UNREGISTERED		
	JXFS_E_REMOTE		

setProperty

Syntax	void setProperty (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:
	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 IJxfsMagStripeControl

4.3.1 Introduction

The J/XFS MSD Device Control Subclass is defined in JxfsMagStripe and is a subclass of JxfsBaseControl. Its interface is defined in IJxfsMagStripeControl interface which is a subclass of IJxfsBaseControl interface. The purpose of the J/XFS MSD 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.

Summary

Although IJxfsMagStripeControl 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 IJxfsMagStripeControl consists on the following methods:

- Getters of listed properties.
- Methods listed.

Property	Туре	Access	Initialized after
deviceType	int	R	
mediaStatus	JxfsMediaStatus	R	
supportedReadTracks	JxfsMSDTracks	R	
supportedWriteTracks	JxfsMSDTracks	R	

Method	Return	May be used after
get <i>Property</i>	Property	
readData	identificationID	
writeData	identificationID	

4.3.2 Properties

deviceType Property (R)

	_	
Туре	int	
Initial Value	Depends on device type.	
Description	on Identifies a type of MSD device. Depending on the device	
_	be a combination of the followin	g flags:
	Value	Meaning
	JXFS_MSD_TYPE_SWIPE	Swipe/pull through magnetic
		stripe reader/encoder.
	JXFS_MSD_TYPE_DIP	Dip magnetic card reader/encoder.
	JXFS_MSD_TYPE_MOTOR	Motorized card reader.

mediaStatus Property (R)		
Туре	JxfsMediaStatus	
Initial Value	A JxfsMediaStatus (see related s document).	section in Base Architecture
Description	Specifies the state of the media.	
Event	If the value of this property changes, the Device Service will send all registered StatusListeners a Status Event with the following values:	
	Field	Value
	status	JXFS_S_MSD_MEDIA_STATUS <i>mediaStatus</i> has changed.
	details	A JxfsMediaStatus object.

Page 12 CWA 13937-3:2000

supportedReadTracks Property (R)

Туре	JxfsMSDTracks
Initial Value	Null until open.
Description	Indicates which tracks can be physically read by the device.

supportedWriteTracks Property (R)

Туре	JxfsMSDTracks
Initial Value	Null until open.
Description	Indicates which tracks can be physically written by the device.

4.3.3 Methods

readData Method					
S	Syntax	identificationID re throws JxfsExcepti		(JxfsMSDTrackS	Selection tracksToRead)
I	Description	This method launch the tracks specified			ain the data contained in imeter.
		Otherwise, the devi cancelled. After a successful c	ce waits completio	until it is present on of this input op	
I	Parameter	Type JxfsMSDTracksSel		Name tracksToRead	Meaning Tracks to be read.
I	Event) operation <i>Event</i> event itionCom <i>Value</i> JXFS_C Identific JXFS_R Operation JXFS_E No read tracks sp been read It is poss read. Ch on track JXFS_E Media v JXFS_E No appr JXFS_E	2 MSD_READD 2 MSD_READD 2 MSD_READD 2 C_SUCCESSFU 2 CANCELLED 3 MSD_READFA 3 CONCELLED 3 MSD_READFA 3 CONCELLED 4 CONDITIONS WERE 4 CONDITIONS WERE 4 CONDITIONS WERE 5 CONCELLED 5 MSD_READFA 5 CONCELLED 5 MSD_READFA 5 CONCELLED 5 CONCE	by MSD Device Control to will contain the data ATA blete operation. L cessfully. AILURE satisfied (that is, not all <i>ToRead</i> parameter have hat some tracks could be for extended information HA re operation completion. DMEDIA ras found.

			D_SHUTTERFAIL not be opened.
	data	A JxfsMSD	ReadData object.
	IntermediateEver	nt	·
	IntermediateEvent	can be sent by	y MSD Device Control to all
	registered Interme	diateListeners	
	Field	Val	ue
	operationID		S_O_MSD_READDATA
	identificationID		tification ID of operation.
	reason		S_I_MSD_NO_MEDIA_PRESENT
			read operation request cannot
			gress because there is no media
			rted.
			S_I_MSD_MEDIA_INSERTED
			read operation request continues
	7.		use a media has been inserted.
	data	null	
Exceptions			<i>ue codes.</i> See section on eption value codes.
	Value		Meaning
	JXFS_E_MSD_N	OTSUPPORT	1
	EDTRACK		tracksToRead parameter is not
	IVEG E MOD N		supported by the device.
	JXFS_E_MSD_N	JIKACKS	No tracks specified in
			tracksToRead parameter.

writeData Method

Syntax	identificationID write throws JxfsException	v	ctor wdata, boolean newCard)
Description	This method initiates a	a write operation of	of the data contained in w <i>data</i> .
	Otherwise, the device	waits until it is pro meter <i>newCard</i> co	is performed immediately. esent or the operation is ontains <i>true</i> , the card must be
		nent 0 contains da] with the data to be written in ta for track 1, vector element 1
	If no data has to be wr element has to contain	U	ack, the corresponding vector
	After a successful con <i>OperationCompleteEv</i> the results.		put operation, an to inform the application of
Parameter	Type java.util.Vector	Name wdata	Meaning Data to be written. Each vector element contains a byte [] of raw data per track. A null vector element is assumed no data to be written for its associated track.

	boolean	newC	Card	If false, it specifies that the operation may proceed when a card is already present.
Event	OperationCompl When a writeData OperationComple all registered Ope Field operationID identificationID result	a () operation is teteEvent event w rationComplete Value JXFS_O_MS	vill be sent Listeners. D_WRITE Id of com	by MSD Device Control to CDATA plete operation.
		Operation con JXFS_E_CAI Operation wa was a card alu true. JXFS_E_MSI No write cond It is possible, written. Chec information o JXFS_E_MSI	npleted suc NCELLED s cancelled ready prese D_WRITE ditions wer however, t k <i>data</i> obje n tracks ac D_NOMEI moved beff D_INVAL ted media v D_BADDA d. D_MEDIA ned.	ccessfully. by application or there ent and <i>waitCard</i> value was FAILURE e satisfied. that some tracks could be ect for extended tually written. DIA fore operation completion. IDMEDIA was found. ATA JAMMED
		Shutter could	-	
	data IntermediateEve	A JxfsMSDT	racks obje	ect.
	IntermediateEvent can be sent by MSD Device		vice Control to all	
	registered Interme			
	Field	Valu	ie	
	operationID identificationID reason	Iden JXF The prog inser JXF The	tification Id S_I_MSD_ write opera ress becaus ted. S_I_MSD_ write opera	_WRITEDATA d of operation. NO_MEDIA_PRESENT ation request cannot se there is no media MEDIA_INSERTED ation request continues a has been inserted.
	data	null		
Exceptions	Some possible Jxf JxfsExceptions fo Value JXFS_E_MSD_N EDTRACK JXFS_E_MSD_N	r other JxfsExce	eption valu Meaning At least o is not suj	e codes.

4.4 IJxfsChipCardControl

4.4.1 Introduction

The J/XFS Chip Card Device Control Subclass is defined in JxfsChipCard and is a subclass of JxfsDeviceControl. Its interface is defined in IJjxfsCCDControl interface which is a subclass of IJxfsBaseControl interface. The purpose of the J/XFS CCD 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.

This class represents a physical device (or part of it) that has chip card access capabilities (send/receive of commands and data).

Summary

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

Therefore, the IJxfsChipCardControl consists on the following methods:

- Getters of listed properties.
- Methods listed.

Property	Туре	Access	Initialized after
deviceType	int	R	
mediaStatus	JxfsMediaStatus	R	

Method	Return	May be used after
get <i>Property</i>	Property	
isCcdT	boolean	
chipInit	identificationID	
chipIO	identificationID	

4.4.2 Properties

deviceType Property (R)

Type Initial Value Description	<i>int</i> Depends on device type. Identifies a type of Chip Card device. Depending on the device type it will be a combination of the following flags:	
	Value JXFS_CCD_TYPE_SWIPE	Meaning Swipe/pull through chip card device.
	JXFS_CCD_TYPE_DIP JXFS_CCD_TYPE_MOTOR JXFS_CCD_TYPE_CONTACT LESS	Dip chip card device. Motorized chip card device. Contactless chip card device.

mediaStatus Property (R)	
Туре	JxfsMediaStatus
Initial Value	A JxfsMediaStatus (see related section in Base Architecture document).
Description	Specifies the state of the media.
Event	If the value of this property changes, the Device Service will send all registered StatusListeners a Status Event with one of the following

values:	
Field	Value
status	JXFS_S_CCD_MEDIA_STATUS
	mediaStatus has changed.
details	A JxfsMediaStatus object.

4.4.3 Methods

isCcdT Method				
	Syntax	boolean isCcdT (int noOfProtocol) throws	s JxfsException;
	Description	supported by the d	protocol Tnn, where nn is	n which protocols are the value of the parameter,
	Parameter	Type int	Name noOfProtocol	Meaning Number of protocol being queried, from 0 to 15 for protocols T0 to T15.
	Exceptions	No additional exce for common value		e section on JxfsExceptions
chipInit Method	Syntax	identificationID	chipInit () throws JxfsExe	ception;
	Description	Performs a chip ca data.	ard initialization and reads	the answer to reset (ATR)
			t, the operation is perform vice waits until it is presen	
			completion of this operati teEvent event is issued to	
	Event	event will be sent	operation is completed an by CCD Device Control to teListeners. It will contain Value JXFS_O_CCD_CHIPIN Identification Id of comp JXFS_RC_SUCCESSFU Operation completed suc Check <i>data</i> field for AT JXFS_E_CANCELLED Operation was cancelled JXFS_E_CCD_IOERRO IO error occurred. No A JXFS_E_CCD_NOMEI	the data read. IIT plete operation. UL ccessfully. IR data from chip. I. DR TR data is available. DIA fore operation completion. IDMEDIA was found.

		JXFS_E_CCD_SHUTTERFAIL
		Shutter could not be opened.
	data	A JxfsCCDData object.
		It contains ATR data from chip.
	IntermediateEven	nt
	IntermediateEvent	can be sent by CCD Device Control to all
	registered Interme	diateListeners
	Field	Value
	operationID	JXFS_O_CCD_CHIPINIT
	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	No additional exce	eptions are generated. See section on JxfsExceptions
	for common value	codes.

chipIO Method

Syntax	identificationID cl JxfsException;	hipIO (by	te[] chipData,	int protocol) throws
Description	This method initiates an input/output operation. The contents of <i>chipData</i> is sent to the chip card. Replied data from the chip card is returned to the application in an <i>OperationCompleteEvent</i> event. The parameter <i>protocol</i> specifies the protocol to use.			
	After a successful <i>OperationComplet</i> the results.			ion, a inform the application of
Parameter	Type byte[]		Name chipData	Meaning Data to be sent.
	int		protocol	Protocol to be used (015).
Event	event will	operation l be sent b nCompleta Value JXFS_C Identifica JXFS_F Operation Check a JXFS_E Operation JXFS_E IO erron JXFS_E Media v JXFS_E Media i JXFS_E	by CCD Device eListeners. It wi o_CCD_CHIPIC cation Id of com C_SUCCESSF on completed su lata field for dat C_CANCELLEE on was cancelled c_CCD_IOERR cocurred. No d C_CCD_NOME	plete operation. UL ccessfully. a returned from chip.) d. OR ata is available. DIA fore operation completion IDMEDIA was found. JAMMED ERFAIL

		JXFS_E_CCD_BADDATA	
		Chip reported data was bad.	
		JXFS_E_CCD_BADPROTOCOL	
		Protocol not supported.	
	data	A JxfsCCDData object.	
		It contains data returned from chip if operation	
		completed successfully.	
	IntermediateEve	nt	
	IntermediateEvent	can be sent by CCD Device Control to all	
	registered Interme	diateListeners	
	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	No additional exce	eptions are generated. See section on JxfsExceptions	
	for common value codes.		

4.5 IJxfsMotorizedCard

4.5.1 Introduction

This interface contains those properties and functions commonly supported in motorized card devices (such as motorized magnetic card readers/encoder and chip card stations) related with its mechanical capabilities like eject or retain cards.

It is intended that this interface will be implemented by device controls that represent physical devices able to manage cards with chip or magnetic stripes (that is, subclasses of JxfsMagStripe and JxfsChipCard classes) that are equipped with motorized and mechanical capabilities.

Summary

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

Therefore, the IJxfsMotorizedCard consists on the following methods:

- Getters of listed properties.
- Methods listed.

Property	Туре	Access	Initialized after
powerOffCapabilities	int	R	
powerOnCapabilities	int	R	
retainBinStatus	JxfsThresholdStatus	R	
retainCardCount	int	R	
retainCapability	boolean	R	
secureModuleType	int	R	

Method	Return	May use after
get <i>Property</i>	Property	
setProperty	void	
resetRetainCardCount	void	
ejectCard	identificationID	
retainCard	identificationID	

4.5.2 Properties

powerOffCapabilities Property (R)

Туре	int		
Initial Value	Depends on device.		
Description	Indicates the action taken by the device at power off if media is present. Depending on the device capabilities it will be set with one of the following values:		
	Value	Meaning	
	JXFS_MOTOR_EJECT	Card is ejected.	
	JXFS_MOTOR_EJECT_THEN_RE	Card is ejected, then, after some	
	TAIN	seconds, it is retained.	
	JXFS_MOTOR_NOACTION	No action is taken.	
	JXFS_MOTOR_READ_POSITION	Card is brought to the read/write	
		position.	
	JXFS_MOTOR_RETAIN	Card is retained.	

powerOnCapabilities Property (R)

Type

Initial Value Description	Depends on device. Indicates the action taken by the device Depending on the device capabilities following values:	it will be set with one of the	
	Value	Meaning	
	JXFS_MOTOR_EJECT	Card is ejected.	
	JXFS_MOTOR_EJECT_THEN_RE TAIN	Card is ejected, then, after some seconds, it is retained.	
	JXFS_MOTOR_NOACTION	No action is taken.	
	JXFS_MOTOR_READ_POSITION	Card is brought to the read/write position.	
	JXFS_MOTOR_RETAIN	Card is retained.	
retainBinStatus Property (R)			
Type Initial Value	<i>JxfsThresholdStatus</i> A JxfsThresholdStatus (see related section in Base Architecture		

- 7	0 <i>j</i> =		
Initial Value	A JxfsThresholdStatus (see related section in Base Architecture		
	document).		
Description	Indicates the fill status of the retain bin, if supported.		
Event	If the value of this property changes, the Device Service will send all		
	registered StatusListeners a Status Event with the following value:		
	Field Value		
	status	JXFS_S_MOTOR_BIN_STATUS	
		retainBinStatus has changed.	
	details	A JxfsThresholdStatus object.	

retainCardCount Property (R/W)

Ту	pe	int			
In	itial Value	Depends on device a	Depends on device at open.		
D	escription	Number of cards retained. This value is persistent between power			
	-	failures, open and clo	ose.		
		The resetRetainCard	<i>Count</i> method resets this property to 0.		
Ev	vent	If the value of this pr	operty changes (increments), the Device Service		
		will send all registered	ed StatusListeners a StatusEvent with a status		
		value of:			
		Field	Value		
		status	JXFS_S_MOTOR_BIN_CARDRE		
			TAINED		
			retainCardCount has incremented.		
		details	None.		

retainCapability Property (R)

Type	boolean
Initial Value	Depends on device.
Description	Indicates if device is able to retain cards.
_	True means it is able to retain, false no retain capability support.

secureModuleType Property (R)

Туре	int	
Initial Value	Depends on device.	
Description	Contains the secure modu	le type, if any being used by the device.
_	Value	Meaning

JXFS_MSD_SECTYPE_NOTSU No security module available. PPORTED JXFS_MSD_SECTYPE_MMBO MMBox module. X JXFS_MSD_SECTYPE_CIM86 CIM86 module

4.5.3 Methods

resetRetainCar	dCount Method		
	Syntax	void resetRetainC	ardCount ()
	Description	Sets retainCardCo	
ejectCard Meth	od		
	Syntax	identificationID	ejectCard () throws JxfsException;
	Description	Ejects the card all	owing card taking from user.
	Event		
		OperationCompl	eteEvent
			() operation is completed an
			<i>teEvent</i> event will be sent by the Device Control to
			rationCompleteListeners with the following data:
		Field operationID	Value JXFS_O_MOTOR_EJECTCARD
		identificationID	The corresponding Id.
		result	JXFS_RC_SUCCESSFUL
			Operation completed successfully.
			This implies that the media has been presented.
			JXFS_E_CANCELLED
			Operation was cancelled.
			JXFS_E_MOTOR_MEDIAJAMMED
			Media is jammed.
			JXFS_E_MOTOR_SHUTTERFAIL Shutter could not be opened.
			JXFS_E_MOTOR_NOMEDIA
			There is no media to eject.
		data:	null.
		IntermediateEve	nt
			t can be sent by CCD Device Control to all
		registered Interme	
		Field	Value
		operationID	JXFS_O_MOTOR_EJECTCARD
		<i>identificationID</i>	Identification Id of operation.
		result:	JXFS_I_MOTOR_MEDIATAKEN The media has been taken by the
			user.
		data	null
	Exceptions		eptions are generated. See section on JxfsExceptions
	T .	for common value	

Page 22 CWA 13937-3:2000

retainCard Method

Syntax	identificationID	retainCard () throws JxfsException;	
Description	Retains card.		
Event	OperationComple	<i>rd()</i> operation is completed an <i>eteEvent</i> event will be sent by the Device Control to rationCompleteListeners. Value JXFS_O_MOTOR_RETAINCARD	
Exceptions	<i>data</i> No additional exc for common value	There is no media to retain. null eptions are generated. See section on JxfsExceptions	

4.6 IJxfsMSDSecure

4.6.1 Introduction

This interface contains properties and functions that may be supported in motorized card MSD devices with a security box instaled.

It is intended that this interface will be implemented by device controls that represent physical devices with the security feature..

Summary

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

Therefore, the IJxfsMSDSecure consists on the following methods:

- Getters of listed properties.
- Methods listed.

Property	Туре	Access	Initialized after
secureModuleKey	byte []	R/W	
secureModuleStatus	int	R	

Method	Return	May be used after
getProperty	Property	
setProperty	void	
readData	identificationID	
readWMtrack	identificationID	

4.6.2 Properties

secureModuleKey Property (R/W)

Туре	byte []
Initial Value	null
Description	Contains the secure module key with parity. Its value should be
	introduced once and be kept after power off.

secureModuleStatus Property (R)

Type Initial Value	<i>int</i> Depends on device at open	
	Depends on device at open.	
Description	Indicates the status of the security n	nodule, if any.
	Value	Meaning
	JXFS_S_MSD_SEC_READY	Security module ready.
	JXFS_S_MSD_SEC_NOTREADY	Security module not ready.
	JXFS_S_MSD_SEC_UNKNOWN	State of the security module
		cannot be determined with the
		device in its current state.
Event	If the value of this property changes	, the Device Service will send all
	registered StatusListeners a StatusE	vent with a status value of:
	Field	Value
	status	XFS_S_MSD_SEC_STATUS
		secureModuleStatus has changed.
	details	None.

4.6.3 Methods

readData Method

Syntax	identificationID JxfsMSDSecureN			kSelection tracksToRead, s JxfsException;	
Description	This method over	loads the	normal readData	a method.	
	It launches a read operation to obtain the data contained in the tracks specified by the <i>tracksToRead</i> parameter.				
	Otherwise, the decancelled. After a successful	vice wait complet	s until it is presention of this input of	erformed immediately. Int or the operation is operation, an o inform the application of	
Parameter	Type JxfsMSDTracksS	election	Name tracksToRead	Meaning Tracks to be read.	
	JxfsMSDSecureN	Iode	secureMode	Required settings for secure operation.	
Event	all registered Ope read. Field operationID identificationID result	() operative event of rationConstructionCo	tion is completed event will be sent mpleteListeners. O_MSD_READ ication ID of con RC_SUCCESSF ion completed su E_CANCELLEI ion was cancelled E_MSD_READ d conditions were specified in <i>track</i> ead). ssible, however, check <i>data</i> object ks actually read. E_MSD_NOME was removed bei E_MSD_INVAL propriated media E_MSD_MEDIA is jammed. E_MSD_SHUTT r could not be opt	by MSD Device Control to It will contain the data DATA nplete operation. UL accessfully. D d. FAILURE e satisfied (that is, not all <i>csToRead</i> parameter have that some tracks could be t for extended information DIA fore operation completion LIDMEDIA was found. AJAMMED FERFAIL ened.	
		e nt t can be	•	Secure object. vice Control to all	
	registered Interme Field	enateList	Value		
	operationID identificationID		JXFS_O_MSE	D_READDATA ID of operation.	

	reason data	The read o progress be inserted. JXFS_I_M The read o	ISD_NO_MEDIA_PRESENT peration request cannot ecause there is no media ISD_MEDIA_INSERTED peration request continues media has been inserted.
Exceptions	Some possible JxfsExcepti JxfsExceptions for other Jx Value JXFS_E_MSD_NOTSUP EDTRACK JXFS_E_MSD_NOTRAC	fsException Mea ORT At le <i>tract</i> supp KS No t	value codes. uning east one track specified in <i>ksToRead</i> parameter is not ported by the device. racks specified in
	JXFS_E_MSD_NOTSUPI EDCAP	ORT The	<i>ksToRead</i> parameter. service does not have secure bility.

readWMtrack

Syntax	identificationID	readWMtrack () throws JxfsException;
Description	This method laund the Watermark.	ches a read operation to obtain the data contained in
	-	t, the read operation is performed immediately. vice waits until it is present or the operation is
	<i>OperationComple</i> the results.	completion of this input operation, an <i>eteEvent</i> event is issued to inform the application of
Event	OperationComple	leteEvent () operation is completed an <i>eteEvent</i> event will be sent by MSD Device Control to rationCompleteListeners. It will contain the data
	Field	Value
	operationID	JXFS_O_MSD_READDATA
	identificationID	Identification ID of complete operation.
	result	JXFS_RC_SUCCESSFUL
		Operation completed successfully.
		JXFS_E_CANCELLED
		Operation was cancelled.
		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 JxfsMSDWmData with Watermark data.
Exceptions		fsException value codes. See section on
	JxfsExceptions fo	r other JxfsException value codes.
	Value	Meaning

Page 26 CWA 13937-3:2000

JXFS_E_MSD_NOTSUPPORT Watermark is not supported. EDTRACK

5 Support Classes

5.1 JxfsMSDTracks

This class provides properties and methods to query which tracks of a MSD device have been selected, are active or have been written.

Used by readData method.

Summary

Implements : -- Extends : JxfsType

Property	Туре	Access	Initialized after
track1	boolean	R	
track2	boolean	R	
track3	boolean	R	

Method	Return	May use after
is <i>Property</i>	Property	
allTracks	boolean	
noTracks	boolean	
JxfsMSDTracks (boolean	(constructor of the class)	
track1, boolean track2,		
boolean track3)		

5.1.1 Properties

track1 Property (R)

TypebooleanInitial ValueFALSEDescriptionIndicates if track1 is selected.ValueMeaningFALSETrack1 is not selected.TRUETrack1 is selected.

track2 Property (R)

Type Initial Value Description *boolean* FALSE Indicates if track2 is selected. **Value** FALSE TRUE

Meaning Track2 is not selected. Track2 is selected.

track3 Property (R)

Type Initial Value Description boolean FALSE Indicates if track3 is selected. Value FALSE TRUE

Meaning Track3 is not selected. Track3 is selected. Page 28 CWA 13937-3:2000

5.1.2 Methods

isTrack1 .. isTrack3 Methods

Syntax	boolean isTrack1 () boolean isTrack3 ()
Description	Return TRUE if specific track property is set to TRUE.

allTracks Method

Syntax	boolean allTracks ()
Description	Returns TRUE if all tracks (track1, track2 and track3) are set to
_	TRUE.

noTracks Method

Syntax	boolean noTracks ()
Description	Returns TRUE if all tracks (track1, track2 and track3) are set to
_	FALSE.

JxfsMSDTracks Constructor

Syntax Description *JxfsMSDTracks (boolean track1, boolean track2, boolean track3)* Constructor of the class.

5.2 JxfsMSDTrackSelection

This class provides properties and methods to query and select the active tracks of a MSD device.

Summary

Implements :		Extends :	JxfsMSDTracks
Property	Туре	Access In	nitialized after

No additional properties.

Method	Return	May use after
setProperty	void	
setAllTracks	void	
setNoTracks	void	
JxfsMSDTrackSelection	(constructor of the class)	
(boolean track1, boolean		
track2, boolean track3)		

5.2.1 Properties

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

5.2.2 Methods

setTrack1 setTrack3 Methods	
Syntax	<pre>void setTrack1 () void setTrack3 ()</pre>
Description	Set specific track property to TRUE.

setAllTracks Method		
Syntax	void setAllTracks ()	
Description	Sets all tracks (track1, track2 and track3 properties) to TRUE.	

	Treeles	Mathad
Setino	Tracks	Method

Syntax	void setNoTracks ()
Description	Sets all tracks (<i>track1</i> , <i>track2</i> and <i>track3</i> properties) to FALSE.

JxfsMSDTrackSelection Constructor

Syntax	JxfsMSDTrackSelection (boolean track1, boolean track2, boolean
	track3)
Description	Constructor of the class.

5.3 JxfsMSDReadData

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

Summary

Implements :		Extends	: JxfsType
Property	Туре	Access	Initialized after
dataRead	java.util.Vector	R	
tracksRead	JxfsMSDTracks	R	
resultReadTrack1	int	R	
resultReadTrack2	int	R	
resultReadTrack3	int	R	

Method	Return	May use after
get <i>Property</i>	Property	
JxfsMSDReadData	(constructor of the class)	
(java.util.Vector dataRead,		
JxfsMSDTracks		
tracksRead, int		
resultReadTrack1, int		
resultReadTrack2, int		
resultReadTrack3)		

5.3.1 Properties

dataRead Property (R)	
Туре	java.util.Vector
Description	Vector of three byte []. Each one contains the raw data read from a track. Vector element 0 contains data for track 1, vector element 1 contains data for track 2, and so on. If no data has been read for a given track, the corresponding vector element contains null . The track data has no SS, SE nor BCC included. The possible elements are ASCII characters 'A' to 'F' and '0' to '9'.

tracksRead Property (R)

Туре	JxfsMSDTracks
Description	Indicates which tracks were effectively read.

resultReadTrack1, resultReadTrack2, resultReadTrack3 Properties (R)

Туре	Int
Initial Value	Depends on device type.
Description	Holds the error code resulting from the read operation for the tracks
-	that could not be read. Should be consulted when a global read error
	JXFS_E_MSD_READFAILURE has been reported.
	Applications must not rely on specific error codes since these may
	depend on the specific device for a given faulty card.
	They will be set with one of the following values:
	Value Meaning

JXFS_E_MSD_NOTSUPPORTE	Track not supported by device.
DTRACK	
JXFS_E_MSD_READFAILURE	Read error on track.
JXFS_E_MSD_PARITY	Parity read error.
JXFS_E_MSD_READ_EOF	Only SS,SE,BCC on track.
JXFS_E_MSD_NO_STRIPE	No magnetic stripe or flux on stripe
	detected (if device has capability to
	detect this situation).
JXFS_E_MSD_READ_OTHER	Any other type of error.

5.3.2 Methods

JxfsMSDReadData Constructor

Syntax	JxfsMSDReadData (java.util.Vector dataRead, JxfsMSDTracks tracksRead, int resultReadTrack1, int resultReadTrack2, int resultReadTrack3)
Description	Constructor of the class.

5.4 JxfsCCDData

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

Summary

Implements :		Extends	: JxfsType
Property	Туре	Access	Initialized after
chipData	byte[]	R	

Method	Return	May use after
get <i>Property</i>	Property	
JxfsCCDData (byte[]	(constructor of the class)	
chipData)		

5.4.1 Properties

chipData	Property (R)	
----------	--------------	--

Туре	byte[]
Description	Contains the data returned by the chip card after a successfull completion of an I/O operation.
	If operation completed is <i>chipInit()</i> , then it contains the ATR data from the chip.
	If operation completed is <i>chipIO()</i> , then it contains the data replied by the chip.

5.4.2 Methods

JxfsCCDData Constructor

Syntax	JxfsCCDData (byte[] chipData)
Description	Constructor of the class.

5.5 JxfsMSDWmData

This class contains the data returned by an *OperationCompleteEvent* event for *readWMtrack()* operation.

Summary

Implements :	 Extends :	JxfsType	

Property	Туре	Access	Initialized after
wmData	byte[]	R	

Method	Return	May use after
getProperty	Property	
JxfsMSDWmData (byte[]	(constructor of the class)	
wmData)		

5.5.1 Properties

wmData Property (R)	
Type	<i>byte[]</i>
Description	Contains the raw Watermark data read

5.5.2 Methods

JxfsMSDWmData Constructor

Syntax	JxfsMSDWmData (byte[] wmData)
Description	Constructor of the class.

5.6 JxfsMSDSecureMode

This class provides required properties for *readData()* operation in secure mode.

Summary

Implements :	Extends :	J	xfsType
Property	Туре	Access	Initialized after
securityCheck	boolean	R/W	
secureTestCard	boolean	R/W	

Method	Return	May be used after
is <i>Property</i>	Property	
setProperty	void	
JxfsMSDSecureMode	(constructor of the class)	
(boolean securityCheck,		
boolean secureTestCard)		

5.6.1 Properties

securityCheck Property (R/W)

Туре	boolean	
Initial Value	FALSE	
Description	Indicates whether a security check has to be requested in read operation. Since the overloaded method will normally be used when security check is desired, this property will usually be TRUE.	
	Value TRUE FALSE	Meaning Security check requested. No security check requested.

securityTestCard Property (R/W)

Туре	boolean	
Initial Value	FALSE	
Description	Indicates whether the	e card to be read is an ecCard or a Test Card.
	Value	Meaning
	TRUE	Test card to be read.
	FALSE	Normal card to be read.

5.6.2 Methods

JxfsMSDSecureMode Constructor

Syntax	JxfsMSDSecureMode (boolean securityCheck, boolean secureTestCard)
Description	Constructor of the class.

5.7 **JxfsMSDReadDataSecure**

This class contains the data returned by an *OperationCompleteEvent* event for *readData()* method in secure mode.

Summary

Implements :	Extends : JxfsType		
Property	Туре	Access	Initialized after
readData	JxfsMSDReadData	R	
securityInfo	int	R	
testResult	byte	R	
cim86Info	byte [8]	R	

Method	Return	May be used after
get <i>Property</i>	Property	
JxfsMSDReadDataSecure	(constructor of the class)	
(JxfsMSDReadData		
readData, int securityInfo,		
byte testResult, byte[]		
cim86Info)		

5.7.1 Properties

readData Property (R)

Туре	JxfsMSDReadData	
Description	This class contains the data obtained from <i>readData()</i> operation as the unsecure mode. <i>See JxfsMSDReadData</i> class for details.	
securityInfo Property (R)		
Туре	int	

could be one of the following value	ies:
Value	Meaning
JXFS_MSD_SEC_NOCHECK	No security check was requeste
JXFS_MSD_SEC_NOTREADY	Security module was not ready
JXFS_MSD_SEC_SECFAIL	Security module failed reading media security sign.
JXFS_MSD_SEC_SECOK	Successful security check.

testResult Property (R)

Туре Description byte

Holds the test result for a given test card. See CIM-86 specifications.

cim86Info Property (R)

Туре	byte[8]
Description	Contains detailed result of the security check in the read operation for
	CIM-86 modules. See CIM-86 specifications.

Page 36 CWA 13937-3:2000

5.7.2 Methods

JxfsMSDReadDataSecure Constructor

Syntax	JxfsMSDReadDataSecure (JxfsMSDReadData readData, int
	securityInfo, byte testResult, byte[] cim86Info)
Description	Constructor of the class.

6 Codes

6.1 Error Codes

Value	Meaning
JXFS_E_MSD_READFAILURE	No read conditions were satisfied (that is, not all
	tracks specified in <i>tracksToRead</i> parameter have
	been read or no Watermark was read).
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.
JXFS_E_MSD_NOTSUPPORTED	At least one track specified in tracksToRead
TRACK	parameter is not supported by the device.
JXFS_E_MSD_NOTRACKS	No tracks specified in <i>tracksToRead</i> parameter.
JXFS_E_MSD_WRITEFAILURE	No write conditions were satisfied.
JXFS_E_MSD_BADDATA	Data is invalid.
JXFS_E_MSD_NOTSUPPORTED	The service does not have secure capability.
CAP	
JXFS_E_MSD_PARITY	Parity read error.
JXFS_E_MSD_READ_EOF	Only SS,SE,BCC on track.
JXFS_E_MSD_NO_STRIPE	No magnetic stripe or flux on stripe detected (if
	device has capability to detect this situation).
JXFS_E_MSD_READ_OTHER	Any other type of read error.

Value	Meaning
JXFS_E_CCD_IOERROR	IO error occurred. No ATR data is available.
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.

Value	Meaning
JXFS_E_MOTOR_MEDIAJAMMED	Media is jammed.
JXFS_E_MOTOR_SHUTTERFAIL	Shutter could not be opened.
JXFS_E_MOTOR_NOMEDIA	There is no media to eject.
JXFS_E_MOTOR_BINFULL	Retain bin is full.

6.2 Status Codes

Value	Meaning	
JXFS_S_MSD_MEDIA_STATUS	mediaStatus property has changed.	
Value	Meaning	
JXFS_S_CCD_MEDIA_STATUS	mediaStatus property has changed.	
Value	Meaning	
JXFS_S_MOTOR_BIN_STATUS	retainBinStatus property has changed.	
JXFS_S_MOTOR_BIN_CARDRE	retainCardCount property has incremented.	
TAINED		

Value	Meaning
JXFS_S_MSD_SEC_STATUS	secureModuleStatus property has changed.
JXFS_S_MSD_SEC_READY	Security module ready.
JXFS_S_MSD_SEC_NOTREADY	Security module not ready.
JXFS_S_MSD_SEC_UNKNOWN	State of the security module cannot be determined
	with the device in its current state.

6.3 Operation Codes

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

Value	Method
JXFS_O_MSD_READDATA	readData, readWMtrack
JXFS_O_MSD_WRITEDATA	writeData

Value	Method
JXFS_O_CCD_CHIPINIT	chipInit
JXFS_O_CCD_CHIPIO	chipIO

Value	Method
JXFS_O_MOTOR_EJECTCARD	ejectCard
JXFS_O_MOTOR_RETAINCARD	retainCard

The following codes identify the reason for an IntermediateEvent:

Value	Meaning
JXFS_I_MSD_NO_MEDIA_PRES	The read operation request cannot progress because
ENT	there is no media inserted.
JXFS_I_MSD_MEDIA_INSERTE	The read operation request continues because a
D	media has been inserted.

Meaning
The read operation request cannot progress because
there is no media inserted.
The read operation request continues because a
media has been inserted.

Value	Meaning
JXFS_I_MOTOR_MEDIATAKEN	The media has been taken by the user.

6.4 Constants

Value	Meaning
$JXFS_MSD_TYPE_SWIPE = 1$	Swipe/pull through magnetic stripe reader/encoder.
$JXFS_MSD_TYPE_DIP = 2$	Dip magnetic card reader/encoder.
$JXFS_MSD_TYPE_MOTOR = 4$	Motorized card reader.
JXFS_MSD_SECTYPE_NOTSUP	No security module available.
PORTED	
JXFS_MSD_SECTYPE_MMBOX	MMBox module available.
JXFS_MSD_SECTYPE_CIM86	CIM86 module available.

Value	Meaning
$JXFS_CCD_TYPE_SWIPE = 1$	Swipe/pull through chip card device.
$JXFS_CCD_TYPE_DIP = 2$	Dip chip card device.
JXFS_CCD_TYPE_MOTOR = 4	Motorized chip card device.
JXFS_CCD_TYPE_CONTACTLE	Contactless chip card device.
SS = 8	_

Value	Meaning
JXFS_MOTOR_EJECT	At power off /on card is ejected.
JXFS_MOTOR_EJECT_THEN_R	At power off /on card is ejected, then, after some
ETAIN	seconds, it is retained.
JXFS_MOTOR_NOACTION	At power off /on no action is taken.
JXFS_MOTOR_READ_POSITIO	At power off /on card is brought to the read/write
Ν	position.
JXFS_MOTOR_RETAIN	At power off /on card is retained.

Value	Meaning
JXFS_MSD_SEC_NOCHECK	No security check was requested.
JXFS_MSD_SEC_NOTREADY	Security module was not ready.
JXFS_MSD_SEC_SECFAIL	Security module failed reading media security sign.
JXFS_MSD_SEC_SECOK	Successful security check.