CEN

WORKSHOP

AGREEMENT

CWA 16008-9

August 2009

ICS 35.240.40

English version

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Part 9: Depository Device Class Interface - Programmer's Reference

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

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

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

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

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

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

Management Centre: Avenue Marnix 17, B-1000 Brussels

Ref. No.:CWA 16008-9:2009 E

Contents

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 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 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 Secretariat, and at http://www.cen.eu/cenorm/sectors/sectors/isss/activity/jxfs_membership.asp. The specification was agreed upon by the J/XFS Workshop Meeting of 2009-05-6/9 in Brussels, and the final version was sent to CEN for publication on 2009-06-12.

The specification is continuously reviewed and commented in the CEN J/XFS Workshop. The information published in this CWA is furnished for informational purposes only. CEN makes no warranty expressed or implied, with respect to this document. Updates of the specification will be available from the CEN J/XFS Workshop public web pages pending their integration in a new version of the CWA (see http://www.cen.eu/cenorm/sectors/sectors/isss/activity/jxfs_cwas.asp).

The J/XFS specifications are now further developed in the CEN J/XFS Workshop. CEN Workshops are open to all interested parties offering to contribute. Parties interested in participating and parties wanting to submit questions and comments for the J/XFS specifications, please contact the J/XFS Workshop Secretariat hosted in CEN (jxfs-helpdesk@cen.eu).

Questions and comments can also be submitted to the members of the J/XFS Forum through the J/XFS Forum web-site http://www.jxfs.net.

This CWA is composed of the following parts:

- Part 1: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Base Architecture - Programmer's Reference
- Part 2: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Pin Keypad Device Class Interface - Programmer's Reference
- Part 3: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Magnetic Stripe & Chip Card Device Class Interface - Programmer's Reference
- Part 4: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Text Input/Output Device Class Interface - Programmer's Reference
- Part 5: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Cash Dispenser, Recycler and ATM Device Class Interface - Programmer's Reference
- Part 6: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Printer Device Class Interface - Programmer's Reference
- Part 7: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Alarm Device Class Interface - Programmer's Reference
- Part 8: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Sensors and Indicators Unit Device Class Interface - Programmer's Reference
- Part 9: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Depository Device Class Interface - Programmer's Reference
- Part 10: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Check Reader/Scanner Device Class Interface - Programmer's Reference (deprecated in favour of Part 13)
- Part 11: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Camera Device Class Interface - Programmer's Reference
- Part 12: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Vendor Dependant Mode Specification - Programmer's Reference
- Part 13: J/eXtensions for Financial Services (J/XFS) for the Java Platform – Scanner Device Class Interface - Programmer’s Reference (recommended replacement for Part 10)

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

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

1 History

Main differences to CWA 14923-9:2004 are:

- Improved description how the open method works
- Replaced OperationCompleteEvent by JxfsOperationCompleteEvent and also for StatusEvent

Main differences to CWA 13937-9:2000 are:

- Added method where JXFS_I_DEP_NO_MEDIA_INSERTED occurrs in summary of IJxfsDepositoryControl.
- Specifies error/exception code when null parameter has been passed to entryEnvelope() or retractEnvelope()
- New constants JXFS_DEP_NO_SCAN and JXFS_DEP_NO_TRANSPORT for value 0.
- Renamed methods of JxfsDepTransportDirection class for a better understanding.
- changed data field of OCE of clearTransport() method and introduced JxfsDepTransportDirection class.
- Additional comment for method dispenseEnvelope(). The operation is completed when the envelope has been taken.
- The details property of a JXFS_S_DEP_TRANSPORT event of the dispenseEnvelope() method is of the type JxfsDepUnitStatus
- The details property of a JXFS_S_DEP_SCANNER event of the readImage() method is of the type JxfsDepUnitStatus
- Spelling error in isMotorized() method of JxfsDepEnvSupplyCapability class. JXFS_DEP_ENVMOTORIZED instead of JXFS_DEP_MOTORIZED.
- New method setNumOfDeposits in interface IJxfsDepositoryControl because the number may be stored in the hardware and therefore setting this value requires hardware access that should be asynchroneous.
- introduced resetRetractCount() method. This method was mentioned in the description of the JxfsDepRetractCount class.
- Solved inconsistencies in the class JxfsDepUnitStatus .
- New comment for JxfsDepReadImageCapability and JxfsDepTransportCapability classes that their property may have the value zero, if none of the specified capabilities is available.
- Added JXFS_S_DEP_RETRACTCOUNT to the list of the status events in the "Events" chapter.
- New coment in description of entryEnvelope() method what has to be done after a JXFS_I_DEP_ENVTAKEN event.
- Comment that a null reference as String is not allowed for the entryEnvelope() and retractEnvelope() methods.
- Renamed all references of JxfsDepImageData to JxfsDepImage.
- In the method readImage() the desired image format may be optionally selected.
- The flag combination of JxfsDepPrintCapability has been reformulated and the isPrinterWithRibbon() method has been added.
- The summary of the events of the IJxfsDepositoryControl interface was not complete
- Extended the description of the clearTransport method.

2 Scope

This document describes the depository device class 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 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 depository devices the basic Device Control structure is extended with various properties and methods specific to this device which are described on the following pages.

3 Overview

3.1 Description

A Depository is used for the acceptance and deposit of media into the device or terminal. There are two main types of depository supported by the J/XFS Depository Device Class: An envelope depository for the deposit of media in envelopes and a night safe depository for the deposit of bags containing bulk media.

- envelope depository for the deposit of media in envelopes
  An envelope depository accepts media, prints on the media and deposits the media into a holding container or bin. Some envelope depositories offer the capability to dispense an envelope to the customer at the start of a transaction. The customer takes this envelope, fills in the deposit media, possibly inscribes it and puts it into the deposit slot. The envelope is then accepted, printed and transported into a deposit container.

  The envelope dispense mechanism may be part of the envelope depository device mechanism with the same entry/exit slot or it may be a separate mechanism with separate entry/exit slot.

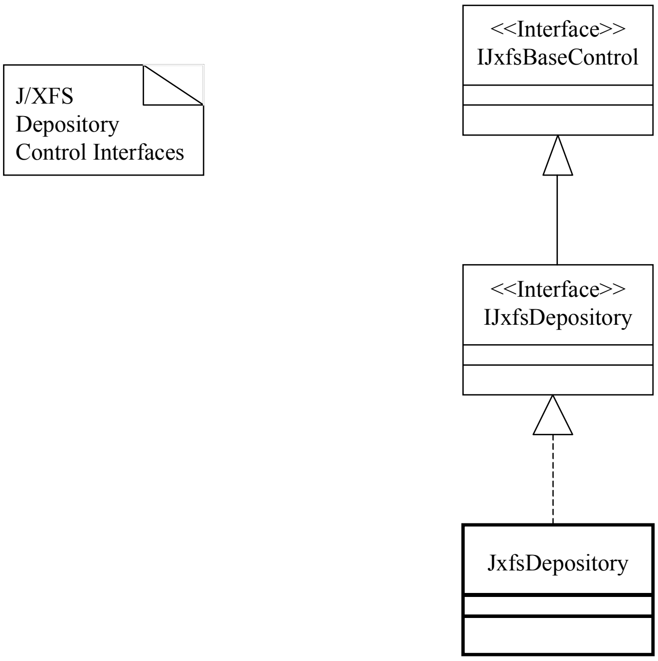
  Envelopes dispensed and not taken by the customer can be retracted back into the device. When the dispenser is a separate mechanism the envelope is retracted back into the dispenser container. When the dispenser is a common mechanism the envelope is retracted into the depository container.

  Some envelope depositories are able to read image data from inserted media.

- night safe depository
  A night safe depository normally only logs the deposit of a bag and does not print on the media.

The J/XFS Depository Device Support uses the event driven model. The application will instantiate a J/XFS Depository Device Control Object and then calls the defined I/O methods with passing data objects containing the parameters. When an I/O method is called, the J/XFS Depository Device Support 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 Depository Device Control Object for the various types of events it wishes to handle.

3.2 Class Hierarchy



3.3 Classes and Interfaces

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

Class or Inter-face	**Name**	**Description**	**Extends / Implements**
Inter-face	**IJxfsBaseControl**	Base interface for all device controls. Contains methods specific to all the device controls.	--
Class	**JxfsBaseControl**	Base class for all device controls. Implements the methods defined in the *IJxfsBaseControl* interface. Contains the properties specific to all device controls.	Implements: IJxfsBaseControl
Inter-face	**IJxfsDepositoryControl**	Base interface for all depository controls. Contains the methods specific to all the device controls for the depository device category.	Extends: IJxfsBaseControl
Class	**JxfsDepository**	Class for the Document Depository control	Extends: JxfsBaseControl Implements: IJxfsDepositoryControl
Inter-face	**IJxfsEventNotification**	Includes one callback method per event type. The Device Service calls these methods to cause events to be delivered to the application.	--

4 Classes and Interfaces

All operation methods return an identificationID. If a method cannot be processed immediately a JxfsException is thrown.

After processing has taken place, a *JxfsOperationCompleteEvent* is generated which contains detailed information about the status of the operation, i.e. if it failed or succeeded, and
if applicable 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 read or write a property the application must use the appropriate methods as defined in the JavaBeans specification.

get*Property*

Syntax	**Property** ***get*****Property*****(void) throws JxfsException;***
Description	Returns the requested property.
Parameter	**None**
Event	No additional events are generated.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes. JXFS_E_CLOSED JXFS_E_REMOTE JXFS_E_UNREGISTERED

set*Property*

Syntax	**Property** ***set*****Property*****(void) throws JxfsException;***
Description	Sets the requested property.
Parameter	**None**
Event	No additional events are generated.
Exceptions	Some possible JxfsException *value codes*. See section on JxfsExceptions for other JxfsException value codes. JXFS_E_CLOSED JXFS_E_PARAMETER_INVALID JXFS_E_REMOTE JXFS_E_UNREGISTERED

4.2 IJxfsDepositoryControl

The J/XFS Depository Device Control is defined in JxfsDepository. Its interface is defined in *IJxfsDepositoryControl*which extends IJxfsBaseControl. The intent of the J/XFS Depository Device Control object is to allow data and control to pass between the application and the device support code so that the associated device can be accessed.

4.2.1 Summary

Property	Type	Access
entryCapability	**JxfsDepEntryCapability**	R
envSupplyCapability	**JxfsDepEnvSupplyCapability**	R
numOfDeposits	**JxfsDepNumOfDeposits**	R/W
printCapability	**JxfsDepPrintCapability**	R
retractCount	**JxfsDepRetractCount**	R/W
retractCapability	**JxfsDepRetractCapability**	R
status	**JxfsDepStatus**	R
readImageCapability	**JxfsDepReadImageCapability**	R
transportCapability	**JxfsDepTransportCapability**	R

Method	Return
get*Property*	*Property*
set*Property*	*Property*
clearTransport	identificationID
dispenseEnvelope	identificationID
entryEnvelope	identificationID
readImage	identificationID
retractEnvelope	identificationID
setNumOfDeposits	identificationID
resetRetractCount	identificationID

Event	May occur after
JxfsStatusEvent	
JXFS_S_DEP_CONTAINER	*clearTransport, entryEnvelope(), retractEnvelope(),readImage()*
JXFS_S_DEP_DEVICE	*clearTransport(), dispenseEnvelope(), entryEnvelope(), retractEnvelope(), readImage()*
JXFS_S_DEP_ENVDISPENSER	*dispenseEnvelope()*
JXFS_S_DEP_ENVSUPPLY	*dispenseEnvelope()*
JXFS_S_DEP_PRINTER	*entryEnvelope(), retractEnvelope()*
JXFS_S_DEP_SCANNER	*readImage()*
JXFS_S_DEP_SHUTTER	*clearTransport(), dispenseEnvelope(), entryEnvelope(), retractEnvelope(), readImage()*
JXFS_S_DEP_TONER	*entryEnvelope(), retractEnvelope()*
JXFS_S_DEP_TRANSPORT	*clearTransport(), dispenseEnvelope(), entryEnvelope(), retractEnvelope(), readImage()*
JxfsIntermediateEvent	
JXFS_E_DEP_DEPOSITERROR	*entryEnvelope()*
JXFS_I_DEP_ENVDEPOSITED	*entryEnvelope()*
JXFS_I_DEP_ENVTAKEN	*entryEnvelope(), dispenseEnvelope()*
JXFS_I_DEP_MEDIA_INSERTED	*readImage()*
JXFS_I_DEP_NO_MEDIA_PRESENT	*readImage()*

Event	May occur after
JxfsOperationCompleteEvent	
JXFS_O_DEP_CLEAR_TRANSPORT	*clearTransport()*
JXFS_O_DEP_DISPENSE_ENVELOPE	*dispenseEnvelope()*
JXFS_O_DEP_ENTRY_ENVELOPE	*entryEnvelope()*
JXFS_O_DEP_READ_IMAGE	*readImage()*
JXFS_O_DEP_RETRACT_ENVELOPE	*retractEnvelope()*
JXFS_O_DEP_SET_NUM_OF_DEPOSITS	*setNumOfDeposits()*
JXFS_O_DEP_RESET_RETRACT_COUNT	*resetRetractCount()*

4.2.2 Properties

status (R)

Type ***JxfsDepStatus***

Initial Value *a JxfsDepStatus* (for initial values see *JxfsDepStatus*)

Description see *JxfsDepStatus*

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with one of the following status values. The values of the data field of the status event object are as follows:

Value	Data Field
JXFS_S_DEP_CONTAINER	JxfsDepRUnitStatus
JXFS_S_DEP_DEVICE	JxfsStatus
JXFS_S_DEP_ENVDISPENSER	JxfsDepUnitStatus
JXFS_S_DEP_ENVSUPPLY	JxfsDepRUnitStatus
JXFS_S_DEP_PRINTER	JxfsDepUnitStatus
JXFS_S_DEP_SCANNER	JxfsDepUnitStatus
JXFS_S_DEP_SHUTTER	JxfsDepShutterStatus
JXFS_S_DEP_TONER	JxfsThresholdStatus
JXFS_S_DEP_TRANSPORT	JxfsDepUnitStatus
JXFS_S_DEP_RETRACTCOUNT	JxfsDepRetractCount

For the other properties see description of the support classes.

4.2.3 Methods

clearTransport

Syntax ***identificationID clearTransport() throws JxfsException;***

Description This command is used to clear the envelope deposit transport from any envelopes or items left in the entry slot of the device. The envelopes can be either captured (retracted into the box) or completely ejected (transported into exit position).

Parameter **None**

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When a *clearTransport()* operation is completed a JxfsOperationCompleteEvent will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the following data:

Field	**Value**
operationID	JXFS_O_DEP_CLEAR_TRANSPORT
identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	JxfsDepTransportDirection

JxfsStatusEvent

When the status of the deposit container changes a *JxfsStatusEvent* is sent to all registered JxfsStatusEventListeners with the following data:

Field	**Value**
status	JXFS_S_DEP_CONTAINER
details	JxfsDepRUnitStatus

JxfsStatusEvent

When the status of the transport shutter changes a *JxfsStatusEvent* is sent to all registered *JxfsStatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_SHUTTER
details	JxfsDepShutterStatus

JxfsStatusEvent

When the status of the deposit transport unit changes a *JxfsStatusEvent* is sent to all registered *JxfsStatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_TRANSPORT
details	JxfsDepUnitStatus

dispenseEnvelope

Syntax ***identificationID dispenseEnvelope() throws JxfsException;***

Description This command is used to dispense an envelope from the envelope supply. This command will either action the dispensing of an envelope from the envelope supply or will unlock the envelope supply for manual access.

Parameter **None**

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When a *dispenseEnvelope()* operation is completed a *JxfsOperationCompleteEvent* will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the following data:

The operation is successfully completed after the envelope has been taken.

Field	**Value**
operationID	JXFS_O_DEP_DISPENSE_ENVELOPE
identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	none

JxfsIntermediateEvent

If the envelope has been taken by the user the J/XFS Depository Device Control will send a *JxfsIntermediateEvent* to all registered IntermediateListeners with the following data:

Field	**Value**
operationID	JXFS_O_DEP_DISPENSE_ENVELOPE
identificationID	The corresponding ID
reason	JXFS_I_DEP_ENVTAKEN
data	none

JxfsStatusEvent

When the status of the envelope dispenser changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_ENVDISPENSER
details	JxfsDepUnitStatus

JxfsStatusEvent

When the status of the envelope supply changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_ENVSUPPLY
details	JxfsDepRUnitStatus

JxfsStatusEvent

When the status of the transport shutter changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_SHUTTER
details	JxfsDepShutterStatus

JxfsStatusEvent

When the status of the deposit transport unit changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	Value
status	JXFS_S_DEP_TRANSPORT
details	*JxfsDepUnitStatus*

entryEnvelope

Syntax ***identificationID entryEnvelope(java.lang.String printData) throws JxfsException;***

Description This command starts the entry of an envelope and deposits it into the deposit container. If the envelope entered has an incorrect size and the deposit was not completed, the envelope is returned to the exit slot for removal by the customer. A JXFS_I_DEP_ENVTAKEN is sent when the envelope is removed. In this case the operation continues waiting for another envelope until a correct envelope has been entered or the operation is cancelled or there is an error.

If a deposit takes place then this command will report a successful operation and any errors detected during the operation will be returned by the JXFS_E_DEP_DEPOSITERROR event.

Parameter

Type	Name	Meaning
java.lang.String	printData	Specifies the data that will be printed on the envelope that is entered by the customer. A null reference is not allowed and results in a JXFS_E_PARAMETER_INVALID exception or error. If nothing shall be printed, it must be an empty String object.

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When an *entryEnvelope()* operation is completed a *JxfsOperationCompleteEvent* will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the following data: In the case of the JXFS_E_DEP_PTRFAIL result the envelope will still have been successfully deposited.

Field	Value
operationID	JXFS_O_DEP_ENTRY_ENVELOPE
identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	none

JxfsIntermediateEvent

If the envelope has been deposited in the deposit container the J/XFS Depository Device Control will send a *JxfsIntermediateEvent* to all registered IntermediateListeners with the following data:

Field	Value
operationID	JXFS_O_DEP_ENTRY_ENVELOPE
identificationID	The corresponding ID
reason	JXFS_I_DEP_ENVDEPOSITED
data	none

JxfsIntermediateEvent
If an error occured during the deposit operation the J/XFS Depository Device Control will send a *JxfsIntermediateEvent* to all registered IntermediateListeners with the following data:

Field	**Value**
operationID	JXFS_O_DEP_ENTRY_ENVELOPE
identificationID	The corresponding ID
reason	JXFS_E_DEP_DEPOSITERROR
data	int errorCode

JxfsIntermediateEvent
If the envelope has been taken by the user the J/XFS Depository Device Control will send a *JxfsIntermediateEvent* to all registered *JxfsIntermediateListeners* with the following data:

Field	**Value**
operationID	JXFS_O_DEP_ENTRY_ENVELOPE
identificationID	The corresponding ID
reason	JXFS_I_DEP_ENVTAKEN
data	none

JxfsStatusEvent
When the status of the deposit container changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_CONTAINER
details	JxfsDepRUnitStatus

JxfsStatusEvent
When the status of the depository's printer changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_PRINTER
details	JxfsDepUnitStatus

JxfsStatusEvent
When the status of the transport shutter changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_SHUTTER
details	JxfsDepShutterStatus

JxfsStatusEvent
When the status of the toner supply changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_TONER
details	JxfsThresholdStatus

JxfsStatusEvent
When the status of the deposit transport unit changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_TRANSPORT
details	JxfsDepUnitStatus

readImage

Syntax ***identificationID readImage(int source) throws JxfsException;***

Description This command is used to return image data from the current document. If no media is present, the device waits endlessly for media to be inserted. If the device supports several formats for returning image data the format used will be selected by the Device Service / hardware if not specified otherwise and the caller must query the JxfsDepImage instance returned to see which has been used.

Parameter

Type	Name	Meaning
int	source	Specifies from which side of the media the returned image will be as one of the following values: JXFS_DEP_CODELINE JXFS_DEP_IMAGEBACK JXFS_DEP_IMAGEFRONT Optionally this value may be 'or'ed with one of the following values. This is relevant, if the Device Service supports more than one graphic format and the application needs a specified format. JXFS_DEP_IMAGEBMP JXFS_DEP_IMAGEMTF JXFS_DEP_IMAGETIF

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When a *readImage()* operation is completed a *JxfsOperationCompleteEvent* will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the image that have been read.

identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	JxfsDepImage

JxfsIntermediateEvent

If media is inserted and the operation can continue the J/XFS Depository Device Control will send a *JxfsIntermediateEvent* to all registered *IntermediateListeners* with the following data:

operationID	JXFS_O_DEP_READ_IMAGE
identificationID	The corresponding ID
reason	JXFS_I_DEP_MEDIA_INSERTED
data	none

JxfsIntermediateEvent

If no media is present the J/XFS Depository Device Control will send a *JxfsIntermediateEvent* to all registered IntermediateListeners with the following data:

operationID	JXFS_O_DEP_READ_IMAGE
identificationID	The corresponding ID
reason	JXFS_I_DEP_NO_MEDIA_PRESENT
data	none

JxfsStatusEvent

When the status of the deposit container changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_CONTAINER
details	JxfsDepRUnitStatus

JxfsStatusEvent

When the status of the scanner changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_SCANNER
details	JxfsDepUnitStatus

JxfsStatusEvent

When the status of the transport shutter changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_SHUTTER
details	JxfsDepShutterStatus

JxfsStatusEvent

When the status of the deposit transport unit changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	**Value**
status	JXFS_S_DEP_TRANSPORT
details	JxfsDepUnitStatus

retractEnvelope

Syntax ***identificationID retractEnvelope(java.lang.String printData) throws JxfsException;***

Description This command is used to retract an envelope that was not taken by a customer after a *dispenseEnvelope()* operation. The given string is printed on the envelope and the envelope is retracted into the deposit container or back to the envelope dispenser, depending on the capabilities of the physical device.

Parameter

Type	Name	Meaning
java.lang.String	printData	Specifies the data that will be printed on the envelope that is retracted. A null reference is not allowed and results in a JXFS_E_PARAMETER_INVALID exception or error. If nothing shall be printed, it must be an empty String object.

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When a *retractEnvelope()* operation is completed a *JxfsOperationCompleteEvent* will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the following data: If the result of the operation is JXFS_E_DEP_PTRFAIL the envelope will still have been successfully deposited.

Field	Value
operationID	JXFS_O_DEP_RETRACT_ENVELOPE
identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	none

JxfsStatusEvent

When the status of the deposit container changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	Value
status	JXFS_S_DEP_CONTAINER
details	JxfsDepRUnitStatus

JxfsStatusEvent

When the status of the depository's printer changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	Value
status	JXFS_S_DEP_PRINTER
details	JxfsDepUnitStatus

JxfsStatusEvent

When the status of the transport shutter changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	Value
status	JXFS_S_DEP_SHUTTER
details	JxfsDepShutterStatus

JxfsStatusEvent

When the status of the toner supply changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	Value
status	JXFS_S_DEP_TONER
details	JxfsThresholdStatus

JxfsStatusEvent

When the status of the deposit transport unit changes a *JxfsStatusEvent* is sent to all registered *StatusEventListeners* with the following data:

Field	Value
status	JXFS_S_DEP_TRANSPORT
details	JxfsDepUnitStatus

setNumOfDeposits

Syntax ***identificationID setNumOfDeposits(int numOfDeposits) throws JxfsException;***

Description This command is used to set the number of actual deposits in the depository.

Parameter

Type	Name	Meaning
int	numOfDeposits	Specifies the number of deposits to be set. This must be a positive value.

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When a *setNumOfDeposits()* operation is completed a *JxfsOperationCompleteEvent* will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the following data.

Field	Value
operationID	JXFS_O_DEP_SET_NUM_OF_DEPOSITS
identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	none

resetRetractCount

Syntax ***identificationID resetRetractCount() throws JxfsException;***

Description This command is used to set the number of retracted envelopes in the depository to zero.

Events Additional Events can be generated:

JxfsOperationCompleteEvent

When a *resetRetractCount()* operation is completed a *JxfsOperationCompleteEvent* will be sent by J/XFS Depository Device Control to all registered JxfsOperationCompleteListeners with the following data.

Field	Value
operationID	JXFS_O_DEP_RESET_RETRACT_COUNT
identificationID	The corresponding ID
result	Common or device dependent error code. (See section on *Error* Codes).
data	none

5 Support Classes

5.1 Summary

Class	Description
JxfsDepEntryCapability	Specifies the entry capability of the depository device.
JxfsDepEnvSupplyCapability	Specifies type of envelope supply unit.
JxfsDepImage	Specifies the image data returned by the *readImage* method
JxfsDepNumOfDeposits	Specifies the actual number of deposits.
JxfsDepPrintCapability	Specifies the description of a field.
JxfsDepRetractCount	Specifies retract count.
JxfsDepRetractCapability	Specifies the retract capabilities of the depository device.
JxfsDepReadImageCapability	Specifies the read image capabilities of the depository device.
JxfsDepTransportCapability	Specifies the transport capabilities of the depository device.
JxfsDepTransportDirection	Specifies the transport direction an envelope has been transported to.

5.2 JxfsDepEntryCapability

This class specifies the entry capabilities of the depository.

5.2.1 Summary

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

Property	Type	Access
entryCapability	int	R

Constructor	Parameter	Parameter-Type
JxfsDepEntryCapability	entryCapability	int

Method	Return	May be used after
get*Property*	*Property*	
isEnvelopeEntrySupported	boolean	
isBagDropSupported	boolean	

Event	May occur after
none	

5.2.2 Properties

entryCapability (R)

Type *int*

Initial Value 0

Description Specifies the entry capability of the depository device as a combination of the following flags:
JXFS_DEP_ENVELOPE
JXFS_DEP_BAGDROP

5.2.3 Methods

isEnvelopeEntrySupported

Syntax ***boolean isEnvelopeEntrySupported(void);***

Description Returns true if the depository accepts envelopes (the *entryCapability* property contains the value JXFS_DEP_ENVELOPE).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isBagDropSupported

Syntax ***boolean isBagDropSupported(void);***

Description Returns TRUE if the depository accepts bag drops (the *entryCapability* property contains the value JXFS_DEP_BAGDROP).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

5.3 JxfsDepEnvSupplyCapability

This class specifies the envelope supply capabilities of the depository.

5.3.1 Summary

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

Property	Type	Access
envSupplyCapability	int	R

Constructor	Parameter	Parameter-Type
JxfsDepEnvSupplyCapability	envSupplyCapability	int

Method	Return	May be used after
get*Property*	*Property*	
isManual	boolean	
isMotorized	boolean	
isNone	boolean	

Event	May occur after
none	

5.3.2 Properties

envSupplyCapability (R)

Type *int*

Initial Value 0

Description Defines what type of envelope supply unit exists as one of the following flags:
JXFS_DEP_ENVMANUAL
JXFS_DEP_ENVMOTORIZED
JXFS_DEP_ENVNONE

5.3.3 Methods

isManual

Syntax ***boolean isManual(void);***

Description Returns TRUE if the envelope supply is manual and must be unlocked to allow envelopes to be taken. The Intermediate Event JXFS_I_DEP_ENVTAKEN cannot be sent and the *retractEnvelope* method cannot be supported (the *envSupplyCapability* property contains the value JXFS_DEP_ENVMANUAL).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isMotorized

Syntax ***boolean isMotorized(void);***

Description Returns TRUE if the Envelope Supply can dispense envelopes (the *envSupplyCapability* property contains the value JXFS_DEP_ENVMOTORIZED).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isNone

Syntax	***boolean isNone(void);***
Description	Returns TRUE if no Envelope Supply exists or Envelope Supply is manual and envelopes can be taken at any time. The Intermediate Event JXFS_I_DEP_ENVTAKEN cannot be sent and the *retractEnvelope* method cannot be supported (the *envSupplyCapability* property contains the value JXFS_DEP_ENVNONE).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

5.4 JxfsDepImage

This class specifies the data of the image read by the *readImage* method.

5.4.1 Summary

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

Property	Type	Access
imageData	byte[]	R
imageType	int	R

Constructor	Parameter	Parameter-Type
JxfsDepImage	imageData	byte[]
	imageType	int

Method	Return	May be used after
get*Property*	*Property*	

Event	May occur after
none	

5.4.2 Properties

imageData (R)

Type ***byte[]***

Initial Value empty byte[]

Description Image data from the current media.

imageType (R)

Type ***int***

Initial Value 0

Description Set to the image data format and can be one of the following values:
JXFS_DEP_CODELINEDATA
JXFS_DEP_IMAGEBMP
JXFS_DEP_IMAGEMTF
JXFS_DEP_IMAGETIF

5.5 JxfsDepNumOfDeposits

This class defines the actual number of deposits in the depository.

The setActualNumOfDeposits() method is deprecated. The method setNumOfDeposits() of the base Depository interface should be used instead.

5.5.1 Summary

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

Property	**Type**	**Access**
actualNumOfDeposits	int	R/W

Constructor	**Parameter**	**Parameter-Type**
JxfsDepNumOfDeposits	actualNumOfDeposits	int

Method	**Return**	**May be used after**
get*Property*	*Property*	
set*Property*		

Event	**May occur after**
none	

5.5.2 Properties

actualNumOfDeposits (R/W)

Type ***int***

Initial Value 0

Description Specifies the number of envelopes or bags in the deposit container. This value is persistent, i.e. maintained through power failures, opens, closes and system resets.

5.5.3 Methods

No additional methods.

5.6 JxfsDepPrintCapability

This class specifies the print capabilities of the depository.

5.6.1 Summary

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

Property	Type	Access
printCapability	int	R
maxNumOfChars	int	R

Constructor	Parameter	Parameter-Type
JxfsDepPrintCapability	printCapability	int
	maxNumOfChars	int

Method	Return	May be used after
get*Property*	*Property*	
isPrinterAvailable	boolean	
isPrintOnRetractSupported	boolean	
isPrinterWithToner	boolean	
isPrinterWithRibbon	boolean	

Event	May occur after
none	

5.6.2 Properties

printCapability (R)

Type ***int***

Initial Value 0

Description Specifies whether a printer is available, whether it has a toner (or ink) cassette and whether it is capable of printing to retracted envelopes. Allowed are the value JXFS_DEP_PTRNONE (meaning that there is no printer at all) or a combination of t the following flags. A combination that includes JXFS_DEP_PTRRIBBON and JXFS_DEP_PTRTONER is not allowed as it is not logical that one printer has a ribbon and a toner/ink capability.

JXFS_DEP_PTRPRINTONRETRACT
JXFS_DEP_PTRRIBBON
JXFS_DEP_PTRTONER

maxNumOfChars (R)

Type ***int***

Initial Value 0

Description Specifies the maximum number of characters that can be printed on the envelope.

5.6.3 Methods

isPrinterAvailable

Syntax ***boolean isPrinterAvailable(void);***

Description Returns TRUE if a printer is available (the *printerCapability* property has a value other than JXFS_DEP_PTRNONE).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isPrinterWithToner

Syntax ***boolean isPrinterWithToner(void);***

Description Returns TRUE if a printer is available and has a toner (or ink) cassette (the *printCapability* property contains the value JXFS_DEP_PTRTONER).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isPrinterWithRibbon

Syntax ***boolean isPrinterWithRibbon(void);***

Description Returns TRUE if a printer is available and has a ribbon printing technology (the *printCapability* property contains the value JXFS_DEP_PTRRIBBON).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isPrintOnRetractSupported

Syntax ***boolean isPrintOnRetractSupported(void);***

Description Returns TRUE if the device can print on retracted envelopes and is either TRUE or FALSE (the *printCapability* property contains the value JXFS_DEP_PTRPRINTONRETRACT).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

5.7 JxfsDepRetractCount

This class specifies the number of envelopes the depository has retracted.

The setRetractCount() method is deprecated. Use the method resetRetractCount() of the base Depository interface instead.

5.7.1 Summary

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

Property	Type	Access
retractCount	int	R/W

Constructor	Parameter	Parameter-Type
JxfsDepRetractCount	retractCount	int

Method	Return	May be used after
get*Property*	*Property*	
set*Property*		

Event	May occur after
none	

5.7.2 Properties

retractCount (R/W)

Type ***int***

Initial Value 0

Description The number of envelopes retracted. This value is persistent: It is reset to zero by the resetRetractCount method.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	Meaning
JXFS_S_DEP_RETRACT COUNT	*retractCount* changed.

5.7.3 Methods

No additional methods.

5.8 JxfsDepRetractCapability

This class specifies the retract capabilities of the depository.

5.8.1 Summary

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

Property	Type	Access
retractCapability	int	R

Constructor	Parameter	Parameter-Type
JxfsDepRetractCapability	retractCapability	int

Method	Return	May be used after
get*Property*	*Property*	
isRetractToDeposit	boolean	

Event	May occur after
none	

5.8.2 Properties

retractCapability (R)

Type ***int***

Initial Value 0

Description Specifies retract capability of the depository as one of the following values:
JXFS_DEP_RETRACTTODEP
JXFS_DEP_RETRACTTOENV

5.8.3 Methods

isRetractToDeposit

Syntax ***boolean isRetractToDeposit(void);***

Description Returns TRUE, if envelopes are retracted back to the deposit container (the *retractCapability* property contains the value JXFS_DEP_RETRACTTODEP) and FALSE if envelopes are retracted back to the envelope dispenser (the *retractCapability* property contains the value JXFS_DEP_RETRACTTOENV).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

5.9 JxfsDepReadImageCapability

This class specifies the read image capabilities of the depository.

5.9.1 Summary

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

Property	Type	Access
readImageCapability	int	R

Constructor	Parameter	Parameter-Type
JxfsDepReadImageCapability	readImageCapability	int

Method	Return	May be used after
get*Property*	*Property*	
isCodeLineSupported	boolean	
isImageBackSupported	boolean	
isImageBMPSupported	boolean	
isImageBothSupported	boolean	
isImageFrontSupported	boolean	
isImageMTFSupported	boolean	
isImageTIFSupported	boolean	
isScanningAvailable	boolean	

Event	May occur after
none	

5.9.2 Properties

readImageCapability (R)

Type *int*

Initial Value 0

Description Specifies whether the device can scan image data from the media. Depending on the device capability *readImageCapability* will be set as a combination of the following values. If no scanning capability is available, the value is JXFS_DEP_NO_SCAN.

Value	Meaning
JXFS_DEP_CODELINE	Device has capability to read MICR format.
JXFS_DEP_IMAGEBACK	The document will be scanned from the back.
JXFS_DEP_IMAGEBMP	Device has capability to read bmp format.
JXFS_DEP_IMAGEBOTH	The document will be scanned from both sides.
JXFS_DEP_IMAGEFRONT	The document will be scanned from the front.
JXFS_DEP_IMAGEMTF	Device has capability to read mtf format.
JXFS_DEP_IMAGETIF	Device has capability to read tif format.

5.9.3 Methods

isCodeLineSupported

Syntax	***boolean isCodeLineSupported(void);***
Description	Returns TRUE if the device has the capability to read MICR format (the *readImageCapability* property contains the value JXFS_DEP_CODELINE).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isImageBackSupported

Syntax	***boolean isImageBackSupported(void);***
Description	Returns TRUE if the device has the capability to read data from the back of the media (the *readImageCapability* property contains the value JXFS_DEP_IMAGEBACK).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isImageBMPSupported

Syntax	***boolean isImageBMPSupported(void);***
Description	Returns TRUE if the device has the capability to read BMP format (the *readImageCapability* property contains the value JXFS_DEP_IMAGEBMP).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isImageBothSupported

Syntax	***boolean isImageBothSupported(void);***
Description	Returns TRUE if the device has the capability to read data from both sides of the media (the *readImageCapability* property contains the value JXFS_DEP_IMAGEBOTH).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isImageFrontSupported

Syntax	***boolean isImageFrontSupported(void);***
Description	Returns TRUE if the device has the capability to read data from the front of the media (the *readImageCapability* property contains the value JXFS_DEP_IMAGEFRONT).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isImageMTFSupported

Syntax	***boolean isImageMTFSupported(void);***
Description	Returns TRUE if the device has the capability to read MTF format (the *readImageCapability* property contains the value JXFS_DEP_IMAGEMTF).
Parameter	**None**
Exceptions	**None**.
Event	No additional events are generated.

isImageTIFSupported

Syntax	***boolean isImageTIFSupported(void);***
Description	Returns TRUE if the device has the capability to read tif format (the *readImageCapability* property contains the value JXFS_DEP_IMAGETIF).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isScanningAvailable

Syntax	***boolean isScanningAvailable(void);***
Description	Returns TRUE if the device has the capability to scan image data from media (the *readImageCapability* property contains a value other than 0).
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

5.10 JxfsDepTransportCapability

This class specifies the transport capabilities of the depository.

5.10.1 Summary

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

Property	Type	Access
transportCapability	int	R

Constructor	Parameter	Parameter-Type
JxfsDepTransportCapability	transportCapability	int

Method	Return	May be used after
get*Property*	*Property*	
isTransportAvailable	boolean	
isShutterAvailable	boolean	

Event	May occur after
none	

5.10.2 Properties

transportCapability (R)

Type *int*

Initial Value 0

Description Specifies transport capability of the depository as a combination of the following flags:

JXFS_DEP_TRANSPORT
JXFS_DEP_SHUTTER

If no transport capability is available the value of this property is JXFS_DEP_NO_TRANSPORT.

5.10.3 Methods

isTransportAvailable

Syntax ***boolean isTransportAvailable(void);***

Description Returns TRUE if a deposit transport mechanism is available (the *transportCapability* property contains the value JXFS_DEP_TRANSPORT).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

isShutterAvailable

Syntax ***boolean isShutterAvailable(void);***

Description Returns TRUE if a deposit transport shutter is available (the *transportCapability* property contains the value JXFS_DEP_SHUTTER).

Parameter **None**

Exceptions **None**

Event No additional events are generated.

5.11 JxfsDepTransportDirection

This class specifies the direction where an envelope has been transported to.

5.11.1 Summary

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

Property	Type	Access
transportDirection	int	R

Constructor	Parameter	Parameter-Type
JxfsDepTransportDirection	transportDirection	int

Method	Return
get*Property*	*Property*
isEjected	boolean
isRetracted	boolean
isTransportedToUnknown	boolean
isNoEnvelope	boolean
isJammed	boolean

Event	May occur after
none	

5.11.2 Properties

transportDirection (R)

Type ***int***

Description Specifies transport direction of the envelope as one of the following flags:
JXFS_DEP_EJECTED
JXFS_DEP_RETRACTED
JXFS_DEP_TRANSPORTED_TO_UNKNOWN
JXFS_DEP_NO_ENVELOPE
JXFS_DEP_JAMMED

5.11.3 Methods

isEjected

Syntax	***boolean isEjected(void);***
Description	Returns TRUE if the enveloped has been ejected completely.
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isRetracted

Syntax	***boolean isRetracted(void);***
Description	Returns TRUE if the enveloped has been transported into the device.
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isTransportedToUnknown

Syntax	***boolean isTransportedToUnknown(void);***
Description	Returns TRUE if the direction where the enveloped has been transported to is unknown.
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isNoEnvelope

Syntax	***boolean isNoEnvelope(void);***
Description	Returns TRUE if there was no envelope to be moved.
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

isJammed

Syntax	***boolean isJammed(void);***
Description	Returns TRUE if the enveloped could not be moved because it is still jammed.
Parameter	**None**
Exceptions	**None**
Event	No additional events are generated.

6 Enum Classes

6.1 JxfsDEPStatusSelectorEnum

This enumeration class is used for the base getStatus(java.util.List) method.

Extends	Implements
JxfsStatusSelectorEnum	

Field	Returned Type	Description
depStatus	*JxfsDepStatus*	Status of the depository device.
numOfDeposits	*JxfsDepNumOfDeposits*	Number of deposits.
retractCount	*JxfsDepRetractCount*	Number of retracted envelopes.
mediaStatus	*JxfsMediaStatus*	Status of the current media.

7 Status Event Classes

If a device status changes one of the following classes is returned via a Jxfs*StatusEvent*.
This *xxxStatus*-Class is passed with the *details* property of the Jxfs*StatusEvent*.
Each *xxxStatus*-Class provides several methods to query the changed device status.

7.1 JxfsThresholdStatus

This class specifies the status of the toner supply in the depository device.

7.1.1 Summary

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

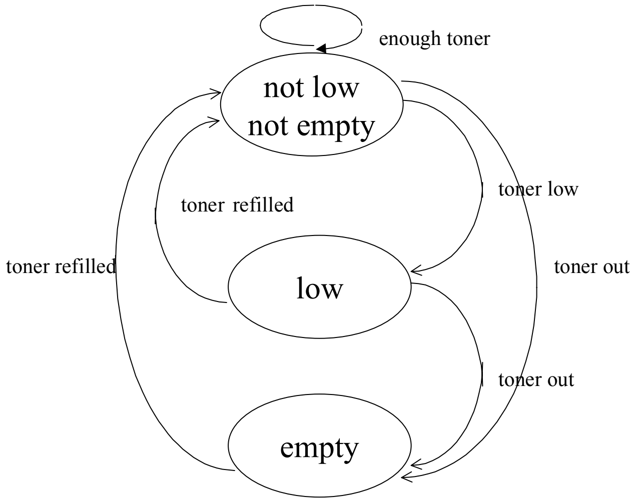
Property	Type	Access
thresholdState	int	R

Constructor	Parameter	Parameter-Type
JxfsThresholdStatus	thresholdState	int

Method	Return	May be used after
isEmpty	*boolean*	
isFull	*boolean*	
isHigh	*boolean*	
isLow	*boolean*	
isUnknown	*boolean*	
toString	*java.lang.String*	

For a description of the class and its properties and methods see "Base Architecture Guide".

The threshold state of the toner supply may change according to the following state transition diagram:



7.2 JxfsDepRUnitStatus

This class specifies the status of a removable unit in the depository device. This can be the deposit container or the envelope supply.

7.2.1 Summary

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

Property	Type	Access
removableUnitStatus	int	R

Constructor	Parameter	Parameter-Type
JxfsDepRUnitStatus	removableUnitStatus	int

Method	Return	May be used after
isEmpty	*boolean*	
isFull	*boolean*	
isHigh	*boolean*	
isInOp	*boolean*	
isLow	*boolean*	
isMissing	*boolean*	
isNotSupported	*boolean*	
isOK	*boolean*	
isUnknown	*boolean*	
isUnlocked	*boolean*	

7.2.2 Properties

removableUnitStatus (R)

Type *int*

Initial Value see Values below

Description Specifies the status of one of the removable units of the depository.

There is a difference in the meaning of this status depending upon the kind of removable unit being referred to. For the envelope supply unit, the status low and empty means, that it is in a state that is not ok (as the envelope supply unit is a unit that is emptied during the operation). For the envelope container unit, the status high and full means, that is is in a state that is not ok (as the envelope container unit is a unit that is filled during the operation). JXFS_S_DEP_OK really means that there is no threshold situation and no error situation.

Depending on device capability, *removableUnitStatus* will be set to one of the following values:

Value	**Meaning**
JXFS_S_DEP_EMPTY	The removable supply unit is present but empty.
JXFS_S_DEP_FULL	The removable container is full.
JXFS_S_DEP_HIGH	The removable container is almost full (threshold).
JXFS_S_DEP_INOP	The removable container or supply unit is in an inoperable state.
JXFS_S_DEP_LOW	The removable supply unit is almost empty (threshold)
JXFS_S_DEP_MISSING	The removable container or supply unit is missing.
JXFS_S_DEP_ NOTSUPPORTED	The removable unit is not installed.
JXFS_S_DEP_OK	The removable supply or container unit is in a good state (and locked).
JXFS_S_DEP_UNKNOWN	State of the removable container or supply unit cannot be determined with the depository in its current state.
JXFS_S_DEP_UNLOCKED	The removable supply or container unit is unlocked.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_RUNIT	*removableUnitStatus* changed.

7.2.3 Methods

isEmpty

Syntax ***boolean isEmpty(void);***

Description Returns TRUE if the removable supply unit is empty (the value of the *removableUnitStatus* property is JXFS_S_DEP_EMPTY).

Exceptions **None**

Event No additional events are generated.

isFull

Syntax ***boolean isFull(void);***

Description Returns TRUE if the removable container unit is full (the value of the *removableUnitStatus* property is JXFS_S_DEP_FULL).

Exceptions **None**

Event No additional events are generated.

isHigh

Syntax ***boolean isHigh(void);***

Description Returns TRUE if the removable container unit is almost full (the value of the *removableUnitStatus* property is JXFS_S_DEP_HIGH).

Exceptions **None**

Event No additional events are generated.

isInOp

Syntax ***boolean isInOp(void);***

Description Returns TRUE if the removable container or supply unit is inoperable (the value of the *removableUnitStatus* property is JXFS_S_DEP_INOP).

Exceptions **None**

Event No additional events are generated.

isLow

Syntax ***boolean isLow(void);***

Description Returns TRUE if the removable container unit is almost empty (the value of the *removableUnitStatus* property is JXFS_S_DEP_LOW).

Exceptions **None**

Event No additional events are generated.

isMissing

Syntax ***boolean isMissing(void);***

Description Returns TRUE if the removable container or supply unit is missing (the value of the *removableUnitStatus* property is JXFS_S_DEP_MISSING).

Exceptions **None**

Event No additional events are generated.

isNotSupported

Syntax	***boolean isNotSupported(void);***
Description	Returns TRUE if the removable unit is not installed (the value of the *removableUnitStatus* property is JXFS_S_DEP_NOTSUPPORTED).
Exceptions	**None**
Event	No additional events are generated.

isOK

Syntax	***boolean isOK(void);***
Description	Returns TRUE if the removable container unit is available and locked (the value of the *removableUnitStatus* property is JXFS_S_DEP_OK).
Exceptions	**None**
Event	No additional events are generated.

isUnknown

Syntax	***boolean isUnknown(void);***
Description	Returns TRUE if the state of the removable container or supply unit cannot be determined with the depository in its current state (the value of the *removableUnitStatus* property is JXFS_S_DEP_UNKNOWN).
Exceptions	**None**
Event	No additional events are generated.

isUnlocked

Syntax	***boolean isUnlocked(void);***
Description	Returns TRUE if the removable supply or container unit is unlocked (the value of the *removableUnitStatus* property is JXFS_S_DEP_UNLOCKED).
Exceptions	**None**
Event	No additional events are generated.

7.3 JxfsDepShutterStatus

This class specifies the status of a transport shutter in the depository device.

7.3.1 Summary

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

Property	Type	Access
shutterStatus	int	R

Constructor	Parameter	Parameter-Type
JxfsDepShutterStatus	shutterStatus	int

Query	Return	May be used after
isShutterClosed	*boolean*	
isShutterJammed	*boolean*	
isShutterOpen	*boolean*	
isShutterNotSupported	*boolean*	
isShutterUnknown	*boolean*	

7.3.2 Properties

shutterStatus (R)

Type ***int***

Description Specifies the status of the transport shutter of the depository. Depending on device capability, *shutterStatus* will be set to one of the following values:

Value	**Meaning**
JXFS_S_DEP_SHTCLOSED	The shutter is closed.
JXFS_S_DEP_SHTJAMMED	The shutter is jammed.
JXFS_S_DEP_SHTOPEN	The shutter is open.
JXFS_S_DEP_ NOTSUPPORTED	The shutter is not present.
JXFS_S_DEP_UNKNOWN	Due to a hardware error or other condition, the state of the shutter cannot be determined.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_SHUTTER	*shutterStatus* changed.

7.3.3 Methods

isShutterClosed

Syntax ***boolean isShutterClosed(void);***

Description Returns TRUE if the shutter is closed (the value of the *shutterStatus* property is JXFS_S_DEP_SHTCLOSED).

Exceptions **None**

Event No additional events are generated.

isShutterJammed

Syntax ***boolean isShutterJammed(void);***

Description Returns TRUE if the shutter is jammed (the value of the *shutterStatus* property is JXFS_S_DEP_SHTJAMMED).

Exceptions **None**

Event No additional events are generated.

isShutterOpen

Syntax ***boolean isShutterOpen(void);***

Description Returns TRUE if the shutter is open (the value of the *shutterStatus* property is JXFS_S_DEP_SHTOPEN).

Exceptions **None**

Event No additional events are generated.

isNotSupported

Syntax ***boolean isNotSupported(void);***

Description Returns TRUE if the physical device has no shutter (the value of the *shutterStatus* property is JXFS_S_DEP_NOTSUPPORTED).

Exceptions **None**

Event No additional events are generated.

isUnknown

Syntax ***boolean isUnknown(void);***

Description Returns TRUE if the state of the shutter cannot be determined with the depository in its current state (the value of the *shutterStatus* property is JXFS_S_DEP_UNKNOWN).

Exceptions **None**

Event No additional events are generated.

7.4 JxfsDepUnitStatus

This class specifies the status of a fixed unit in the depository device. This can be the deposit transport mechanism, the printer, the envelope dispenser or the scanner.

7.4.1 Summary

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

Property	Type	Access
unitStatus	int	R

Constructor	Parameter	Parameter-Type
JxfsDepUnitStatus	unitStatus	int

Method	Return	May be used after
isInOp	*boolean*	
isNotSupported	*boolean*	
isOK	*boolean*	
isUnknown	*boolean*	

7.4.2 Properties

unitStatus (R)

Type ***int***

Description Specifies the status of one of the fixed units of the depository. Depending on device capability, *unitStatus* will be set to one of the following values:

Value	Meaning
JXFS_S_DEP_INOP	The unit is present but in an inoperable state.
JXFS_S_DEP_OK	The unit is present and in a good state.
JXFS_S_DEP_ NOTSUPPORTED	The unit is not present.
JXFS_S_DEP_UNKNOWN	State of the unit cannot be determined with the depository in its current state.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	Meaning
JXFS_S_DEP_ENVDISPENSE R	Envelope dispenser status changed
JXFS_S_DEP_PRINTER	Printer status changed.
JXFS_S_DEP_SCANNER	Scanner status changed.
JXFS_S_DEP_TRANSPORT	Transport mechanism status changed.

7.4.3 Methods

isInOp

Syntax ***boolean isInOp(void);***

Description Returns TRUE if the unit is inoperable (the value of the *unitStatus* property is JXFS_S_DEP_INOP).

Exceptions **None**

Event No additional events are generated.

isOK

Syntax	***boolean isOK(void);***
Description	Returns TRUE if the unit is available and in a good state (the value of the *unitStatus* property is JXFS_S_DEP_OK).
Exceptions	**None**
Event	No additional events are generated.

isNotSupported

Syntax	***boolean isNotSupported(void);***
Description	Returns TRUE if the unit is not present (the value of the *unitStatus* property is JXFS_S_DEP_NOTSUPPORTED).
Exceptions	**None**
Event	No additional events are generated.

isUnknown

Syntax	***boolean isUnknown(void);***
Description	Returns TRUE if the state of the unit cannot be determined with the depository in its current state (the value of the *unitStatus* property is JXFS_S_DEP_UNKNOWN).
Exceptions	**None**
Event	No additional events are generated.

7.5 JxfsDepStatus

This class contains properties and methods to query the status of the depository device and its resources.

7.5.1 Summary

Implements: *Serializable* **Extends:** *JxfsStatus*

Property	Type	Access
containerStatus	JxfsDepRUnitStatus	R
envDispenserStatus	JxfsDepUnitStatus	R
envSupplyStatus	JxfsDepRUnitStatus	R
printerStatus	JxfsDepUnitStatus	R
scannerStatus	JxfsDepUnitStatus	R
shutterStatus	JxfsDepShutterStatus	R
tonerStatus	JxfsThresholdStatus	R
transportStatus	JxfsDepUnitStatus	R

Constructor	Parameter	Parameter-Type
JxfsDepStatus	containerStatus	JxfsDepRUnitStatus
	envDispenserStatus	JxfsDepUnitStatus
	envSupplyStatus	JxfsDepRUnitStatus
	printerStatus	JxfsDepUnitStatus
	scannerStatus	JxfsDepUnitStatus
	shutterStatus	JxfsDepShutterStatus
	tonerStatus	JxfsThresholdStatus
	transportStatus	JxfsDepUnitStatus

Method	Return	May be used after
get*Property*	*Property*	

Event	May occur after
none	

7.5.2 Properties

The value of the data field for the events on properties is the value of the property itself i.e. for the JXFS_S_DEP_CONTAINER event the data field would be the containerStatus (of type JxfsDepRUnitStatus).

containerStatus (R)

Type ***JxfsDepRUnitStatus***

Description Specifies the state of the deposit container that contains the deposited envelopes or bags.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	Meaning
JXFS_S_DEP_CONTAINER	*containerStatus* changed.

envDispenserStatus (R)

Type ***JxfsDepUnitStatus***

Description Specifies the state of the envelope dispenser.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	Meaning
JXFS_S_DEP_ENVDISPENSER	*envDispenserStatus* changed.

envSupplyStatus (R)

Type ***JxfsDepRUnitStatus***

Description Specifies the state of the envelope supply unit.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_ENVSUPPLY	*envSupplyStatus* changed.

printerStatus (R)

Type ***JxfsDepUnitStatus***

Description Specifies the status of the printer.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_PRINTER	*printerStatus* changed.

scannerStatus (R)

Type ***JxfsUnitStatus***

Description Specifies the status of the scanner unit.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_SCANNER	*scannerStatus* changed.

shutterStatus (R)

Type ***JxfsDepShutterStatus***

Description Specifies the status of the transport shutter.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_SHUTTER	*shutterStatus* changed.

tonerStatus (R)

Type ***JxfsThresholdStatus***

Description Specifies the status of the toner supply.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_TONER	*tonerStatus* changed.

transportStatus (R)

Type ***JxfsUnitStatus***

Description Specifies the status of the deposit transport unit.

Event If the value of this property changes, the Device Service will send all registered StatusListeners a *JxfsStatusEvent* with a status value of:

Value	**Meaning**
JXFS_S_DEP_TRANSPORT	*transportStatus* changed.

7.5.3 Events

JxfsIntermediateEvent

Interface	***jxfs.events.IJxfsIntermediateListener***		
Method	*intermediateOccurred(JxfsIntermediateEvent e);*		
Remarks	Issued to present intermediate information from the depository to the application. The data field for the intermediate events is null.		
Properties	**Type**	**Name**	**Meaning**
	int	code	The *code* value can be one of the following Codes:
Codes			
	Value		**Meaning**
	JXFS_E_DEP_DEPOSITERROR		An error occurred during the deposit operation. For every error that occurrs a single event is generated.
	JXFS_I_DEP_ENVDEPOSITED		The envelope has been deposited in the deposit container.
	JXFS_I_DEP_ENVTAKEN		The envelope has been taken by the customer.
	JXFS_I_DEP_NO_MEDIA_ PRESENT		No media is present in the device
	JXFS_I_DEP_MEDIA_ INSERTED		Media has been inserted into the device.

JxfsStatusEvent

Interface	***jxfs.events.IJxfsStatusListener***		
Method	*statusOccurred(JxfsStatusEvent e);*		
Remarks	Issued to present status data from the depository to the application		
Properties	**Type**	**Name**	**Meaning**
	int	status	The *status* value can be one of the following Codes:
	JxfsType	details	The type of the details property is dependant on the value of the status property. See "IJxfsDepositoryControl interface, Properties" for a list of the delivered objects.
Codes			
	Value		**Meaning**
	JXFS_S_DEP_DEVICE		Property *deviceStatus* changed.
	JXFS_S_DEP_CONTAINER		Property *containerStatus* changed.
	JXFS_S_DEP_ENVDISPENSER		Property *envDispenserStatus* changed
	JXFS_S_DEP_ENVSUPPLY		Property *envSupplyStatus* changed
	JXFS_S_DEP_PRINTER		Property *printerStatus* changed
	JXFS_S_DEP_SCANNER		Property *scannerStatus* changed
	JXFS_S_DEP_SHUTTER		Property *shutterStatus* changed
	JXFS_S_DEP_TONER		Property *tonerStatus* changed.
	JXFS_S_DEP_TRANSPORT		Property *transportStatus* changed
	JXFS_S_DEP_RETRACTCOUNT		Retract counter value changed.

8 Codes

8.1 Error Codes

Value	Meaning
JXFS_E_DEP_CONT_MISSING	The deposit container is not present.
JXFS_E_DEP_DEPFULL	The deposit container is full.
JXFS_E_DEP_DEPJAMMED	An envelope jam occurred in the deposit transport.
JXFS_E_DEP_ENVEMPTY	There is no envelope in the envelope unit.
JXFS_E_DEP_ENVJAMMED	An envelope jam occurred.
JXFS_E_DEP_ENVSIZE	The envelope entered has an incorrect size.
JXFS_E_DEP_NOCODELINE	No MICR data was available to read.
JXFS_E_DEP_NOENV	No envelope to retract.
JXFS_E_DEP_PTRFAIL	The printer failed.
JXFS_E_DEP_SHTNOTCLOSED	The shutter failed to close.
JXFS_E_DEP_SHTNOTOPENED	The shutter failed to open.
JXFS_E_DEP_DEPUNKNOWN	The result of the deposit is not known. This error code is only returned by the JXFS_E_DEP_DEPOSITERROR event.

8.2 Status Codes

General Status Codes

General Status Codes that specify a value change.

Value	Meaning
JXFS_S_DEP_DEVICE	The status of the depository device has changed.
JXFS_S_DEP_CONTAINER	The status of the depository's container unit has changed.
JXFS_S_DEP_ENVDISPENSER	The status of the depository's envelope dispenser unit has changed..
JXFS_S_DEP_ENVSUPPLY	The status of the depository's envelope supply has changed.
JXFS_S_DEP_PRINTER	The status of the depository's printer has changed.
JXFS_S_DEP_TONER	The status of the depository's toner supply has changed.
JXFS_S_DEP_TRANSPORT	The status of the depository's deposit transport has changed.
JXFS_S_DEP_SHUTTER	The status of the depository's transport shutter has changed.

Bin Status Codes

Defines the status code the paper supply, the toner supply or the retain bin can report.

Value	Meaning
JXFS_S_BIN_EMPTY	Bin is empty.
JXFS_S_BIN_FULL	Bin is full.
JXFS_S_BIN_HIGH	Bin is high.
JXFS_S_BIN_LOW	Bin is low.
JXFS_S_BIN_NOTSUPPORTED	Capability to report the state of the bin is not supported by the device.
JXFS_S_BIN_OK	Bin is available and neither high nor full.
JXFS_S_BIN_UNKNOWN	State of the bin cannot be determined with the device in its current state.

Removable Unit Status Codes

Defines the status the removable units of the depository can report.

Value	Meaning
JXFS_S_DEP_EMPTY	The removable supply unit is present but empty.
JXFS_S_DEP_FULL	The removable container is full.
JXFS_S_DEP_HIGH	The removable container is almost full (threshold).
JXFS_S_DEP_INOP	The removable unit is present but in an inoperable state.
JXFS_S_DEP_LOW	The removable supply unit is almost empty (threshold).
JXFS_S_DEP_MISSING	The removable unit is missing.
JXFS_S_DEP_NOTSUPPORTED	The removable unit is not installed.
JXFS_S_DEP_OK	The removable container or supply unit is present and in a good state.
JXFS_S_DEP_UNKNOWN	State of the removable unit cannot be determined with the depository in its current state.
JXFS_S_DEP_UNLOCKED	The removable supply or container unit is unlocked.

Unit Status Codes

Defines the status the units of the depository can report.

Value	Meaning
JXFS_S_DEP_INOP	The unit is present but in an inoperable state.
JXFS_S_DEP_OK	The unit is present and in a good state.
JXFS_S_DEP_NOTSUPPORTED	The unit is not present.
JXFS_S_DEP_UNKNOWN	State of the unit cannot be determined with the depository in ist current state.

Shutter Status Codes

Defines the status codes that can be reported for the shutter.

Value	Meaning
JXFS_S_DEP_SHTCLOSED	The shutter is closed.
JXFS_S_DEP_SHTJAMMED	The shutter is jammed.
JXFS_S_DEP_SHTOPEN	The shutter is opened.
JXFS_S_DEP_NOTSUPPORTED	The shutter is not present.
JXFS_S_DEP_UNKNOWN	Due to a hardware error or other condition, the state of the shutter cannot be determined.

8.3 Constants

Entry Capability Codes

The entry capability codes are possible values for the entryCapability property. The values can be or'ed.

Value	Meaning
JXFS_DEP_ENVELOPE	Depository accepts envelopes.
JXFS_DEP_BAGDROP	Depository accepts bags.

Envelope Supply Capability Codes

The envelope supply capability codes are possible values for the envSupplyCapability property.

Value	Meaning
JXFS_DEP_ENVMANUAL	Envelope Supply is manual and must be unlocked to allow envelopes to be taken. The Intermediate Event JXFS_I_DEP_ENVTAKEN cannot be sent and the *retractEnvelope* method can not be supported.
JXFS_DEP_ENVMOTORIZED	Envelope Supply can dispense envelopes.
JXFS_DEP_ENVNONE	No Envelope Supply exists or Envelope Supply is manual and envelopes can be taken at any time. The Intermediate Event JXFS_I_DEP_ENVTAKEN cannot be sent and the *retractEnvelope* method can not be supported.

Print Capability Codes

The print capability codes are possible values for the printCapability property.

Value	Meaning
JXFS_DEP_PTRNONE	There is no printer available.
JXFS_DEP_PTRPRINTONRETRACT	The device can print on retracted envelopes. This value can be or'ed with any of the other values
JXFS_DEP_PTRRIBBON	The available printer has a ribbon.
JXFS_DEP_PTRTONER	The available printer has a toner (or ink) cassette.

Read Image Codes

The read image codes are input or output parameter of the *readImage* method. They are also possible values for the readImageCapability property.

Value	Meaning
JXFS_DEP_NO_SCAN	There is no scanning capability available.
JXFS_DEP_CODELINE	The MICR line of the document will be read.
JXFS_DEP_CODELINEDATA	The returned image is MICR data.
JXFS_DEP_IMAGEBACK	The document will be scanned from the back.
JXFS_DEP_IMAGEBMP	The returned image is in BMP format.
JXFS_DEP_IMAGEBOTH	The document will be scanned from both sides.
JXFS_DEP_IMAGEFRONT	The document will be scanned from the front.
JXFS_DEP_IMAGEMTF	The returned image is in MTF format.
JXFS_DEP_IMAGETIF	The returned image is in TIF format.

Retract Capability Codes

The retract capability codes are possible values for the retractCapability property.

Value	Meaning
JXFS_DEP_RETRACTTODEP	Retracted envelopes are put in the deposit container.
JXFS_DEP_RETRACTTOENV	Envelopes are retracted back to the envelope dispenser.

Transport Capability Codes

The transport capability codes are possible values for the transportCapability property. The following transport capability codes can be or'ed.

Value	Meaning
JXFS_DEP_NO_TRANSPORT	There is no deposit transport mechanism available.
JXFS_DEP_TRANSPORT	A deposit transport mechanism is available.
JXFS_DEP_SHUTTER	A deposit transport shutter is available.

Transport Direction Codes

The transport direction codes are possible values for the transportDirection property.

Value	Meaning
JXFS_DEP_EJECTED	The envelope has been ejected completely
JXFS_DEP_RETRACTED	The envelope has been retracted into the device
JXFS_DEP_TRANSPORTED_TO_UNKNOWN	The direction where the envelope has been transported to is unknown.
JXFS_DEP_NO_ENVELOPE	There has not been an envelope to transport.
JXFS_DEP_JAMMED	The envelope was jammed and could not be transported.

8.4 Operation ID Codes

Following codes specify the operation which generated the JxfsOperationCompleteEvent.

Value	Method
JXFS_O_DEP_CLEAR_TRANSPORT	*clearTransport()*
JXFS_O_DEP_DISPENSE_ENVELOPE	*dispenseEnvelope*()
JXFS_O_DEP_ENTRY_ENVELOPE	*entryEnvelope()*
JXFS_O_DEP_READ_IMAGE	*readImage()*
JXFS_O_DEP_RETRACT_ENVELOPE	*retractEnvelope()*
JXFS_O_DEP_SET_NUM_OF_DEPOSITS	*setNumOfDeposits()*
JXFS_O_DEP_RESET_RETRACT_COUNT	*resetRetractCount()*

9 Device Service Interface Methods

The Device Service interface is common to all Device Services of this device type. It is used by the Device Controls to access the functionality of the device. This interface has to be implemented by any J/XFS Device Service.

The device type specific Device Service interface is similar to the Device Control interface. All device specific method calls are extended by an additional parameter (int control_id). This is always added as the last parameter in every operation.

The name of the device service interface is IJxfsDepositoryService. It is extended from IJxfsBaseService.

9.1 Constant Definitions

9.1.1 Status Codes

Constant	Numerical Value
JXFS_S_DEP_DEVICE	11001
JXFS_S_DEP_CONTAINER	11002
JXFS_S_DEP_ENVDISPENSER	11003
JXFS_S_DEP_ENVSUPPLY	11004
JXFS_S_DEP_PRINTER	11005
JXFS_S_DEP_TONER	11006
JXFS_S_DEP_TRANSPORT	11007
JXFS_S_DEP_SHUTTER	11008
JXFS_S_DEP_EMPTY	11009
JXFS_S_DEP_FULL	11010
JXFS_S_DEP_HIGH	11011
JXFS_S_DEP_INOP	11012
JXFS_S_DEP_LOW	11013
JXFS_S_DEP_MISSING	11014
JXFS_S_DEP_NOTSUPPORTED	11015
JXFS_S_DEP_OK	11016
JXFS_S_DEP_UNKNOWN	11017
JXFS_S_DEP_UNLOCKED	11018
JXFS_S_DEP_SHTCLOSED	11019
JXFS_S_DEP_SHTJAMMED	11020
JXFS_S_DEP_SHTOPEN	11021
JXFS_S_DEP_SCANNER	11047
JXFS_S_DEP_RETRACTCOUNT	11048
JXFS_S_DEP_RUNIT	11049
JXFS_S_DEP_MEDIA_STATUS_CHANGE	11051

9.1.2 Operation ID Codes

Constant	Method
JXFS_O_DEP_CLEAR_TRANSPORT	11024
JXFS_O_DEP_DISPENSE_ENVELOPE	11025
JXFS_O_DEP_ENTRY_ENVELOPE	11026
JXFS_O_DEP_READ_IMAGE	11027
JXFS_O_DEP_RETRACT_ENVELOPE	11028
JXFS_O_DEP_SET_NUM_OF_DEPOSITS	11059
JXFS_O_DEP_RESET_RETRACT_COUNT	11060

9.1.3 Intermediate Codes

Constant	Method
JXFS_I_DEP_ENVDEPOSITED	11029
JXFS_E_DEP_DEPOSITERROR	11030
JXFS_I_DEP_ENVTAKEN	11031
JXFS_I_DEP_MEDIA_INSERTED	11032
JXFS_I_DEP_NO_MEDIA_PRESENT	11033

9.1.4 Error Codes

Constant	Numerical Value
JXFS_E_DEP_DEPOSITERROR	11030
JXFS_E_DEP_SHTNOTOPENED	11034
JXFS_E_DEP_CODELINEDATA	11035
JXFS_E_DEP_CONT_MISSING	11036
JXFS_E_DEP_DEPFULL	11037
JXFS_E_DEP_DEPJAMMED	11038
JXFS_E_DEP_SHTNOTCLOSED	11039
JXFS_E_DEP_DEPUNKNOWN	11040
JXFS_E_DEP_ENVEMPTY	11041
JXFS_E_DEP_ENVJAMMED	11042
JXFS_E_DEP_ENVSIZE	11043
JXFS_E_DEP_NOCODELINE	11044
JXFS_E_DEP_NOENV	11045
JXFS_E_DEP_PTRFAIL	11046

9.1.5 General Codes

readImage/readImage Capability Codes

Constant	Numerical Value
JXFS_DEP_CODELINE	512 (1<<9)
JXFS_DEP_IMAGEBACK	2048 (1<<11)
JXFS_DEP_IMAGEFRONT	16384(1<<14)
JXFS_DEP_IMAGEBMP	4096 (1<<12)
JXFS_DEP_IMAGEMTF	32768 (1<<15)
JXFS_DEP_IMAGETIF	65536 (1<<16)
JXFS_DEP_IMAGEBOTH	8192 (1<<13)
JXFS_DEP_NO_SCAN	0

imageType Codes

Constant	Numerical Value
JXFS_DEP_CODELINEDATA	1024 (1<<10)
JXFS_DEP_IMAGEBMP	4096 (1<<12)
JXFS_DEP_IMAGEMTF	32768 (1<<15)
JXFS_DEP_IMAGETIF	65536 (1<<16)

transportDirection Codes

Constant	Numerical Value
JXFS_DEP_EJECTED	11054
JXFS_DEP_RETRACTED	11055
JXFS_DEP_TRANSPORTED_TO_UNKNOWN	11056
JXFS_DEP_NO_ENVELOPE	11057
JXFS_DEP_JAMMED	11058

9.1.6 Capability Codes

entryCapability Codes

Constant	Numerical Value
JXFS_DEP_ENVELOPE	1 (1<<0)
JXFS_DEP_BAGDROP	2 (1<<1)
JXFS_DEP_EJECTONENVWRONGSIZE	4 (1<<2)

envSupplyCapability Codes

Constant	Numerical Value
JXFS_DEP_ENVMANUAL	4 (1<<2)
JXFS_DEP_ENVMOTORIZED	8 (1<<3)
JXFS_DEP_ENVNONE	16 (1<<4)

printCapability Codes

Constant	Numerical Value
JXFS_DEP_PTRNONE	32 (1<<5)
JXFS_DEP_PTRRIBBON	128 (1<<7)
JXFS_DEP_PTRTONER	256 (1<<8)
JXFS_DEP_PTRPRINTONRETRACT	64 (1<<6)

transportCapability Codes

Constant	Numerical Value
JXFS_DEP_NO_TRANSPORT	0
JXFS_DEP_TRANSPORT	131072 (1<<17)
JXFS_DEP_SHUTTER	262144 (1<<18)

retractCapability Codes

Constant	Numerical Value
JXFS_DEP_RETRACTTODEP	11022
JXFS_DEP_RETRACTTOENV	11023