



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

WORKSHOP AGREEMENT

CWA 13937-9

August 2000

ICS 35.240.40

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part
9: Depository 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 13937-9:2000 E

Foreword

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

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

The specification is continuously reviewed and commented in the CEN/ISSS J/XFS Workshop. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this one. The information published in this CWA is furnished for informational purposes only. CEN/ISSS makes no warranty expressed or implied, with respect to this document. Updates of the specification will be available from the CEN/ISSS J/XFS Workshop public web pages pending their integration in a new version of the CWA (see: <http://www.cenorm.be/iss/wkshop/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 (iss@cenorm.be). To submit questions and comments for the J/XFS specifications, please contact the CEN/ISSS Secretariat (iss@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	SCOPE	5
2	OVERVIEW	6
2.1	DESCRIPTION	6
2.2	CLASS HIERARCHY	7
2.3	CLASSES AND INTERFACES	8
3	DEVICE BEHAVIOR	9
3.1	DEVICE OPEN()	9
4	CLASSES AND INTERFACES	10
4.1	ACCESS TO PROPERTIES	10
4.2	EXCEPTIONS	10
4.3	IJXFSDEPOSITORYCONTROL	11
4.3.1	Summary	11
4.3.2	Properties	12
4.3.3	Methods	13
4.3.4	Events	21
5	SUPPORT CLASSES	22
5.2	JXFSDEPENTRYCAPABILITY	23
5.2.1	Summary	23
5.2.2	Properties	23
5.2.3	Methods	23
5.3	JXFSDEPENVSUPPLYCAPABILITY	24
5.3.1	Summary	24
5.3.2	Properties	24
5.3.3	Methods	24
5.4	JXFSDEPIIMAGE	26
5.4.1	Summary	26
5.4.2	Properties	26
5.5	JXFSDEPNUMOFDEPOSITS	27
5.5.1	Summary	27
5.5.2	Properties	27
5.5.3	Methods	27
5.6	JXFSDEPPRINTCAPABILITY	28
5.6.1	Summary	28
5.6.2	Properties	28
5.6.3	Methods	28
5.7	JXFSDEPRETRACTCOUNT	30
5.7.1	Summary	30
5.7.2	Properties	30
5.7.3	Methods	30
5.8	JXFSDEPRETRACTCAPABILITY	31
5.8.1	Summary	31
5.8.2	Properties	31
5.8.3	Methods	31
5.9	JXFSDEPREADIMAGECAPABILITY	32
5.9.1	Summary	32
5.9.2	Properties	32
5.9.3	Methods	33
5.10	JXFSDEPTRANSPORTCAPABILITY	35
5.10.1	Summary	35
5.10.2	Properties	35
5.10.3	Methods	35
6	STATUS EVENT CLASSES	36
6.1	JXFS THRESHOLDSTATUS	36
6.1.1	Summary	36

6.2	JXFSDEPRUNITSTATUS.....	37
6.2.1	Summary	37
6.2.2	Properties.....	37
6.2.3	Methods.....	38
6.3	JXFSDEPSHUTTERSTATUS.....	40
6.3.1	Summary	40
6.3.2	Properties.....	40
6.3.3	Methods.....	40
6.4	JXFSDEPUNITSTATUS	42
6.4.1	Summary	42
6.4.2	Properties.....	42
6.4.3	Methods.....	42
6.5	JXFSDEPSTATUS	44
6.5.1	Summary	44
6.5.2	Properties.....	44
6.5.3	Events.....	46
7	CODES.....	48
7.1	ERROR CODES.....	48
7.2	EXCEPTION CODES.....	49
7.3	STATUS CODES.....	50
7.4	CONSTANTS.....	52
7.5	OPERATION ID CODES	54
8	DEVICE SERVICE INTERFACE METHODS.....	55
9	INDEX.....	56

1 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.

2 Overview

2.1 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.

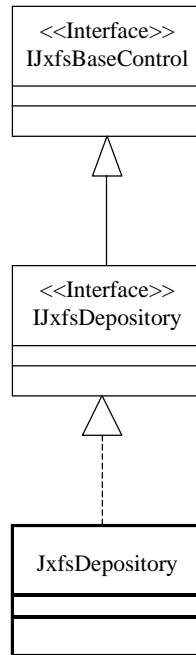
Some envelope depositories are able to read image data from inserted media.

- 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 call 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.

2.2 Class Hierarchy

J/XFS
Depository
Control Interfaces



2.3 Classes and Interfaces

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

Class or Interface	Name	Description	Extends / Implements
Interface	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
Interface	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	Implements: IJxfsDepositoryControl
Interface	IJxfsEventNotification	Includes one callback method per event type. The Device Service calls these methods to cause events to be delivered to the application.	--

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 OutputComplete – Event 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 read or write a property the application must use the appropriate methods as defined in the JavaBeans specification.

4.1.1.1 *getProperty*

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

4.1.1.2 *setProperty*

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

4.2 Exceptions

The methods described for the specific interfaces all can throw at least the following exceptions :

Exception	Value
<i>JxfsException</i>	JXFS_E_CLOSED
	JXFS_E_PARAMETER_INVALID
	JXFS_E_NOT_SUPPORTED
	JXFS_E_REMOTE
	JXFS_E_UNREGISTERED

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

4.3 JxfsDepositoryControl

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.

4.3.1 Summary

Property	Type	Access	Initialized after
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	May be used after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	
clearTransport	identificationID	
dispenseEnvelope	identificationID	
entryEnvelope	identificationID	
readImage	identificationID	
retractEnvelope	identificationID	

Event	May occur after
StatusEvent JXFS_S_DEP_CONTAINER JXFS_S_DEP_DEVICE JXFS_S_DEP_ENVDISPENSER JXFS_S_DEP_ENVSUPPLY JXFS_S_DEP_PRINTER JXFS_S_DEP_SCANNER JXFS_S_DEP_SHUTTER JXFS_S_DEP_TONER JXFS_S_DEP_TRANSPORT	<i>clearTransport(), entryEnvelope(), retractEnvelope(), readImage()</i> <i>clearTransport(), dispenseEnvelope(), entryEnvelope(), retractEnvelope(), readImage()</i> <i>dispenseEnvelope()</i> <i>dispenseEnvelope()</i> <i>entryEnvelope(), retractEnvelope()</i> <i>readImage()</i> <i>clearTransport(), dispenseEnvelope(), entryEnvelope(), retractEnvelope(), readImage()</i> <i>entryEnvelope(), retractEnvelope()</i> <i>clearTransport(), dispenseEnvelope(), entryEnvelope(), retractEnvelope(), readImage()</i>
IntermediateEvent JXFS_E_DEP_DEPOSITERROR JXFS_I_DEP_ENVDEPOSITED JXFS_I_DEP_ENVTAKEN	<i>entryEnvelope()</i> <i>entryEnvelope()</i> <i>entryEnvelope(), dispenseEnvelope()</i>

Event	May occur after
OperationCompleteEvent	
JXFS_O_DEP_CLEAR_TRANSPORT	<i>clearTransport()</i>
JXFS_O_DEP_DISPENSE_ENVELOPE	<i>dispenseEnvelope()</i>
JXFS_O_DEP_ENTRY_ENVELOPE	<i>entryEnvelope()</i>
JXFS_O_DEP_READ_IMAGE	<i>readImage()</i>
JXFS_O_DEP_RETRACT_ENVELOPE	<i>retractEnvelope()</i>

4.3.2 Properties

4.3.2.1 status (R)

Type	<i>JxfsDepStatus</i>
Initial Value	<i>aJxfsDepStatus</i> (for initial values see <i>JxfsDepStatus</i>)
Description	see <i>JxfsDepStatus</i>
Event	If the value of this property changes, the Device Service will send all registered StatusListeners a StatusEvent with one of the following status values :
	Value
	JXFS_S_DEP_CONTAINER
	JXFS_S_DEP_DEVICE
	JXFS_S_DEP_ENVDISPENSER
	JXFS_S_DEP_ENVSUPPLY
	JXFS_S_DEP_PRINTER
	JXFS_S_DEP_SCANNER
	JXFS_S_DEP_SHUTTER
	JXFS_S_DEP_TONER
	JXFS_S_DEP_TRANSPORT

For the other properties see description of the support classes.

4.3.3 Methods

4.3.3.1 clearTransport

Syntax
Description

identificationID clearTransport() throws *JxfsException*;
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 or completely ejected.

Parameter
Exceptions
Events

None
No additional exceptions generated.
Additional Events can be generated :

OperationCompleteEvent

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

Field	Value
<i>operationID</i>	JXFS_O_DEP_CLEAR_TRANSPORT
<i>identificationID</i>	The corresponding ID
<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_DEPFULL JXFS_E_DEP_DEPJAMMED JXFS_E_DEP_CONTMISSING JXFS_E_DEP_SHTNOTCLOSED
<i>data</i>	none

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_CONTAINER
<i>details</i>	JxfsDepRUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_SHUTTER
<i>details</i>	JxfsDepShutterStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TRANSPORT
<i>details</i>	JxfsDepUnitStatus

4.3.3.2 dispenseEnvelope

Syntax
Description

identificationID dispenseEnvelope() throws *JxfsException*;
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
Exceptions
Events

None
No additional exceptions generated.
Additional Events can be generated :

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

Field	Value
<i>operationID</i>	JXFS_O_DEP_DISPENSE_ENVELOPE
<i>identificationID</i>	The corresponding ID
<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_ENVEMPTY JXFS_E_DEP_ENVJAMMED JXFS_E_DEP_SHTNOTOPENED
<i>data</i>	none

IntermediateEvent

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

Field	Value
<i>operationID</i>	JXFS_O_DEP_DISPENSE_ENVELOPE
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_DEP_ENVTAKEN
<i>data</i>	none

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_ENVDISPENSER
<i>details</i>	JxfsDepUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_ENVSUPPLY
<i>details</i>	JxfsDepRUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_SHUTTER
<i>details</i>	JxfsDepShutterStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TRANSPORT
<i>details</i>	JxfsDepTransportStatus

4.3.3.3 entryEnvelope

Syntax	<i>identificationID</i> entryEnvelope (<i>String printData</i>) throws <i>JxfsException</i> ;																														
Description	<p>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.</p> <p>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.</p>																														
Parameter	<table border="0"> <thead> <tr> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td><i>String</i></td> <td>printData</td> <td>Specifies the data that will be printed on the envelope that is entered by the customer.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>String</i>	printData	Specifies the data that will be printed on the envelope that is entered by the customer.																								
Type	Name	Meaning																													
<i>String</i>	printData	Specifies the data that will be printed on the envelope that is entered by the customer.																													
Exceptions	No additional exceptions generated.																														
Events	<p>Additional Events can be generated :</p> <p>OperationCompleteEvent When a <i>entryEnvelope()</i> operation is completed an OperationCompleteEvent will be sent by J/XFS Depository Device Control to all registered OperationCompleteListeners with the following data:</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Field</th> <th style="text-align: left;">Value</th> </tr> </thead> <tbody> <tr> <td><i>operationID</i></td> <td>JXFS_O_DEP_ENTRY_ENVELOPE</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>result</i></td> <td>JXFS_RC_SUCCESSFUL JXFS_E_DEP_ENVJAMMED JXFS_E_DEP_DEPFULL JXFS_E_DEP_CONTMISSING JXFS_E_DEP_ENVSIZE JXFS_E_DEP_PTRFAIL JXFS_E_DEP_SHTNOTCLOSED JXFS_E_DEP_SHTNOTOPENED JXFS_E_DEP_DEPUNKNOWN</td> </tr> <tr> <td><i>data</i></td> <td>none</td> </tr> </tbody> </table> <p>IntermediateEvent If the envelope has been deposited in the deposit container the J/XFS Depository Device Control will send an IntermediateEvent to all registered IntermediateListeners with the following data :</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Field</th> <th style="text-align: left;">Value</th> </tr> </thead> <tbody> <tr> <td><i>operationID</i></td> <td>JXFS_O_DEP_ENTRY_ENVELOPE</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_I_DEP_ENVDEPOSITED</td> </tr> <tr> <td><i>data</i></td> <td>none</td> </tr> </tbody> </table> <p>IntermediateEvent If an error occurred during the deposit operation the J/XFS Depository Device Control will send an IntermediateEvent to all registered IntermediateListeners with the following data :</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Field</th> <th style="text-align: left;">Value</th> </tr> </thead> <tbody> <tr> <td><i>operationID</i></td> <td>JXFS_O_DEP_ENTRY_ENVELOPE</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_E_DEP_DEPOSITERROR</td> </tr> <tr> <td><i>data</i></td> <td>Int errorCode</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE	<i>identificationID</i>	The corresponding ID	<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_ENVJAMMED JXFS_E_DEP_DEPFULL JXFS_E_DEP_CONTMISSING JXFS_E_DEP_ENVSIZE JXFS_E_DEP_PTRFAIL JXFS_E_DEP_SHTNOTCLOSED JXFS_E_DEP_SHTNOTOPENED JXFS_E_DEP_DEPUNKNOWN	<i>data</i>	none	Field	Value	<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_DEP_ENVDEPOSITED	<i>data</i>	none	Field	Value	<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_E_DEP_DEPOSITERROR	<i>data</i>	Int errorCode
Field	Value																														
<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE																														
<i>identificationID</i>	The corresponding ID																														
<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_ENVJAMMED JXFS_E_DEP_DEPFULL JXFS_E_DEP_CONTMISSING JXFS_E_DEP_ENVSIZE JXFS_E_DEP_PTRFAIL JXFS_E_DEP_SHTNOTCLOSED JXFS_E_DEP_SHTNOTOPENED JXFS_E_DEP_DEPUNKNOWN																														
<i>data</i>	none																														
Field	Value																														
<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE																														
<i>identificationID</i>	The corresponding ID																														
<i>reason</i>	JXFS_I_DEP_ENVDEPOSITED																														
<i>data</i>	none																														
Field	Value																														
<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE																														
<i>identificationID</i>	The corresponding ID																														
<i>reason</i>	JXFS_E_DEP_DEPOSITERROR																														
<i>data</i>	Int errorCode																														

IntermediateEvent

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

Field	Value
<i>operationID</i>	JXFS_O_DEP_ENTRY_ENVELOPE
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_DEP_ENVTAKEN
<i>data</i>	none

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_CONTAINER
<i>details</i>	JxfsDepRUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_PRINTER
<i>details</i>	JxfsDepUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_SHUTTER
<i>details</i>	JxfsDepShutterStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TONER
<i>details</i>	JxfsThresholdStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TRANSPORT
<i>details</i>	JxfsDepUnitStatus

4.3.3.4 readImage

Syntax	<i>identificationID readImage(int source) throws JxfsException;</i>																												
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.																												
Parameter	<table border="0"> <thead> <tr> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td><i>int</i></td> <td>source</td> <td>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</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>int</i>	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																						
Type	Name	Meaning																											
<i>int</i>	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																											
Exceptions	No additional exceptions generated.																												
Events	<p>Additional Events can be generated :</p> <p>OCDepReadImageEvent When a <i>readImage()</i> operation is completed an OCDepReadImageEvent will be sent by J/XFS Depository Device Control to all registered OCDepReadImageListeners with the image that have been read.</p> <table border="0"> <tr> <td style="vertical-align: top;"><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td style="vertical-align: top;"><i>result</i></td> <td>JXFS_RC_SUCCESSFUL JXFS_E_DEP_NOCODELINE JXFS_E_DEP_SHTNOTCLOSED JXFS_E_DEP_SHTNOTOPENED</td> </tr> <tr> <td style="vertical-align: top;"><i>data</i></td> <td>JxfsDepImageData</td> </tr> </table> <p>IntermediateEvent If media is inserted and the operation can continue the J/XFS Depository Device Control will send an IntermediateEvent to all registered IntermediateListeners with the following data :</p> <table border="0"> <tr> <td style="vertical-align: top;"><i>operationID</i></td> <td>JXFS_O_DEP_READ_IMAGE</td> </tr> <tr> <td style="vertical-align: top;"><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td style="vertical-align: top;"><i>reason</i></td> <td>JXFS_I_DEP_MEDIA_INSERTED</td> </tr> <tr> <td style="vertical-align: top;"><i>data</i></td> <td>none</td> </tr> </table> <p>IntermediateEvent If no media is present the J/XFS Depository Device Control will send an IntermediateEvent to all registered IntermediateListeners with the following data :</p> <table border="0"> <tr> <td style="vertical-align: top;"><i>operationID</i></td> <td>JXFS_O_DEP_READ_IMAGE</td> </tr> <tr> <td style="vertical-align: top;"><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td style="vertical-align: top;"><i>reason</i></td> <td>JXFS_I_DEP_NO_MEDIA_PRESENT</td> </tr> <tr> <td style="vertical-align: top;"><i>data</i></td> <td>none</td> </tr> </table> <p>StatusEvent When the status of the deposit container changes a StatusEvent is sent to all registered StatusEventListeners with the following data :</p> <table border="0"> <thead> <tr> <th style="text-align: left;">Field</th> <th style="text-align: left;">Value</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"><i>status</i></td> <td>JXFS_S_DEP_CONTAINER</td> </tr> <tr> <td style="vertical-align: top;"><i>details</i></td> <td>JxfsDepRUnitStatus</td> </tr> </tbody> </table>	<i>identificationID</i>	The corresponding ID	<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_NOCODELINE JXFS_E_DEP_SHTNOTCLOSED JXFS_E_DEP_SHTNOTOPENED	<i>data</i>	JxfsDepImageData	<i>operationID</i>	JXFS_O_DEP_READ_IMAGE	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_DEP_MEDIA_INSERTED	<i>data</i>	none	<i>operationID</i>	JXFS_O_DEP_READ_IMAGE	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_DEP_NO_MEDIA_PRESENT	<i>data</i>	none	Field	Value	<i>status</i>	JXFS_S_DEP_CONTAINER	<i>details</i>	JxfsDepRUnitStatus
<i>identificationID</i>	The corresponding ID																												
<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_NOCODELINE JXFS_E_DEP_SHTNOTCLOSED JXFS_E_DEP_SHTNOTOPENED																												
<i>data</i>	JxfsDepImageData																												
<i>operationID</i>	JXFS_O_DEP_READ_IMAGE																												
<i>identificationID</i>	The corresponding ID																												
<i>reason</i>	JXFS_I_DEP_MEDIA_INSERTED																												
<i>data</i>	none																												
<i>operationID</i>	JXFS_O_DEP_READ_IMAGE																												
<i>identificationID</i>	The corresponding ID																												
<i>reason</i>	JXFS_I_DEP_NO_MEDIA_PRESENT																												
<i>data</i>	none																												
Field	Value																												
<i>status</i>	JXFS_S_DEP_CONTAINER																												
<i>details</i>	JxfsDepRUnitStatus																												

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_SCANNER
<i>details</i>	JxfsDepScannerStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_SHUTTER
<i>details</i>	JxfsDepShutterStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TRANSPORT
<i>details</i>	JxfsDepUnitStatus

4.3.3.5 retractEnvelope

Syntax	<i>identificationID retractEnvelope(String printData) throws JxfsException;</i>						
Description	This command is used to retract an envelope that was not taken by a customer after an envelope dispense 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	<table border="0"> <thead> <tr> <th style="text-align: left;">Type</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td><i>String</i></td> <td>printData</td> <td>Specifies the data that will be printed on the envelope that is retracted.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>String</i>	printData	Specifies the data that will be printed on the envelope that is retracted.
Type	Name	Meaning					
<i>String</i>	printData	Specifies the data that will be printed on the envelope that is retracted.					

Exceptions No additional exceptions generated.
Events Additional Events can be generated :

OperationCompleteEvent

When a *retractEnvelope()* operation is completed an OperationCompleteEvent will be sent by J/XFS Depository Device Control to all registered OperationCompleteListeners with the following data:

Field	Value
<i>operationID</i>	JXFS_O_DEP_RETRACT_ENVELOPE
<i>identificationID</i>	The corresponding ID
<i>result</i>	JXFS_RC_SUCCESSFUL JXFS_E_DEP_CONTMISSING JXFS_E_DEP_DEPFULL JXFS_E_DEP_DEPJAMMED JXFS_E_DEP_ENVJAMMED JXFS_E_DEP_NOENV JXFS_E_DEP_PTRFAIL JXFS_E_DEP_SHTNOTCLOSED
<i>data</i>	none

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_CONTAINER
<i>details</i>	JxfsDepRUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_PRINTER
<i>details</i>	JxfsDepUnitStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_SHUTTER
<i>details</i>	JxfsDepShutterStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TONER
<i>details</i>	JxfsThresholdStatus

StatusEvent

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

Field	Value
<i>status</i>	JXFS_S_DEP_TRANSPORT
<i>details</i>	JxfsDepUnitStatus

4.3.4 Events

4.3.4.1 OCDepReadImageEvent

Interface	<i>jxfs.events.OCDepReadImageListener</i>		
Method	OCDepReadImageOccurred(OCDepReadImageEvent <i>e</i>);		
Description	Issued to notify the client that a <i>readImage()</i> command has completed with sending the expected input data.		
Properties	Type	Name	Meaning
	JxfsDepImage	data	Image data from the current media.

5 Support Classes

5.1.1.1 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.

5.2 JxfsDepEntryCapability

This class specifies the entry capabilities of the depository.

5.2.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

Property	Type	Access	Initialized after
entryCapability	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepEntryCapability	entryCapability	int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
isEnvelopeEntrySupported	boolean	
isBagDropSupported	boolean	

Event	May occur after
none	

5.2.2 Properties

5.2.2.1 entryCapability (R)

Type	<i>int</i>
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

5.2.3.1 isEnvelopeEntrySupported

Syntax	<i>boolean isEnvelopeEntrySupported(void);</i>
Description	Returns TRUE if the depository accepts envelopes (the <i>entryCapability</i> property contains the value JXFS_DEP_ENVELOPE).
Parameter	None
Exceptions	None
Event	No additional events are generated.

5.2.3.2 isBagDropSupported

Syntax	<i>boolean isBagDropSupported(void);</i>
Description	Returns TRUE if the depository accepts bag drops (the <i>entryCapability</i> 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	Initialized after
envSupplyCapability	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepEnvSupplyCapability	envSupplyCapability	int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
isManual	boolean	
isMotorized	boolean	
isNone	boolean	

Event	May occur after
none	

5.3.2 Properties

5.3.2.1 envSupplyCapability (R)

Type	<i>int</i>
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

5.3.3.1 isManual

Syntax	<i>boolean isManual(void);</i>
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 <i>retractEnvelope</i> method cannot be supported (the <i>envSupplyCapability</i> property contains the value JXFS_DEP_ENVMANUAL).
Parameter	None
Exceptions	None
Event	No additional events are generated.

5.3.3.2 isMotorized

Syntax	<i>boolean isMotorized(void);</i>
Description	Returns TRUE if the Envelope Supply can dispense envelopes (the <i>envSupplyCapability</i> property contains the value JXFS_DEP_MOTORIZED).
Parameter	None
Exceptions	None
Event	No additional events are generated.

5.3.3.3 isNone

Syntax	<i>boolean isNone(void);</i>
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 <i>retractEnvelope</i> method cannot be supported (the <i>envSupplyCapability</i> 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	Initialized after
imageData	byte[]	R	
imageType	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepImage	imageData	Byte[]
	imageType	Int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	

Event	May occur after
none	

5.4.2 Properties

5.4.2.1 imageData (R)

Type	<i>byte[]</i>
Initial Value	empty byte[]
Description	Image data from the current media.

5.4.2.2 imageType (R)

Type	<i>int</i>
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.

5.5.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

Property	Type	Access	Initialized after
actualNumOfDeposits	int	R/W	

Constructor	Parameter	Parameter-Type
JxfsDepNumOfDeposits	actualNumOfDeposits	int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>		

Event	May occur after
none	

5.5.2 Properties

5.5.2.1 actualNumOfDeposits (R/W)

Type	<i>int</i>
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.
Event	No additional events
Exceptions	Some possible JxfsException <i>value codes</i> . See section on JxfsExceptions for other JxfsException value codes.
Value	Meaning

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	Initialized after
printCapability	int	R	
maxNumOfChars	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepPrintCapability	printCapability	int
	maxNumOfChars	int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
isPrinterAvailable	boolean	
isPrintOnRetractSupported	boolean	
isPrinterWithToner	boolean	

Event	May occur after
none	

5.6.2 Properties

5.6.2.1 printCapability (R)

Type	<i>int</i>
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 as a combination of : JXFS_DEP_PTRPRINTONRETRACT or JXFS_DEP_PTRNONE with on of the following values JXFS_DEP_PTRNONE JXFS_DEP_PTRRIBBON JXFS_DEP_PTRTONER

5.6.2.2 maxNumOfChars (R)

Type	<i>int</i>
Initial Value	0
Description	Specifies the maximum number of characters that can be printed on the envelope.

5.6.3 Methods

5.6.3.1 isPrinterAvailable

Syntax	<i>boolean isPrinterAvailable(void);</i>
Description	Returns TRUE if a printer is available (the <i>printerCapability</i> property has a value other than JXFS_DEP_PTRNONE).
Parameter	None
Exceptions	None
Event	No additional events are generated.

5.6.3.2 isPrinterWithToner

Syntax	<i>boolean isPrinterWithToner(void);</i>
Description	Returns TRUE if a printer is available and has a toner (or ink) cassette (the <i>printCapability</i> property contains the value JXFS_DEP_PTRTONER).
Parameter	None
Exceptions	None
Event	No additional events are generated.

5.6.3.3 isPrintOnRetractSupported

Syntax	<i>boolean isPrintOnRetractSupported(void);</i>
Description	Returns TRUE if the device can print on retracted envelopes and is either TRUE or FALSE. (the <i>printCapability</i> 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.

5.7.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

Property	Type	Access	Initialized after
retractCount	Int	R/W	

Constructor	Parameter	Parameter-Type
JxfsDepRetractCount	retractCount	Int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>		

Event	May occur after
none	

5.7.2 Properties

5.7.2.1 retractCount (R/W)

Type	<i>int</i>
Initial Value	0
Description	The number of envelopes retracted. This value is persistent: It is reset to zero by the <i>resetRetractCount</i> method.
Event	If the value of this property changes, the Device Service will send all registered StatusListeners a StatusEvent with a status value of:
	Value JXFS_S_DEP_RETRACT COUNT
	Meaning <i>retractCount</i> changed.

5.7.3 Methods

No additional methods.

5.8 JxlsDepRetractCapability

This class specifies the retract capabilities of the depository.

5.8.1 Summary

Implements : *Serializable*

Extends : *JxlsType*

Property	Type	Access	Initialized after
retractCapability	Int	R	

Constructor	Parameter	Parameter-Type
JxlsDepRetractCapability	retractCapability	Int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
isRetractToDeposit	Boolean	

Event	May occur after
none	

5.8.2 Properties

5.8.2.1 retractCapability (R)

Type	<i>Int</i>
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

5.8.3.1 isRetractToDeposit

Syntax	<i>boolean isRetractToDeposit(void);</i>
Description	Returns TRUE, if envelopes are retracted back to the deposit container (the <i>retractCapability</i> property contains the value JXFS_DEP_RETRACTTODEP) and FALSE if envelopes are retracted back to the envelope dispenser (the <i>retractCapability</i> 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	Initialized after
readImageCapability	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepReadImageCapability	readImageCapability	int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
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

5.9.2.1 readImageCapability (R)

Type	<i>int</i>
Initial Value	0
Description	Specifies whether the device can scan image data from the media. Depending on the device capability <i>readImageCapability</i> will be set as a combination of the following values:
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

5.9.3.1 isCodeLineSupported

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

5.9.3.2 isImageBackSupported

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

5.9.3.3 isImageBMPSupported

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

5.9.3.4 isImageBothSupported

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

5.9.3.5 isImageFrontSupported

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

5.9.3.6 isImageMTFSupported

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

5.9.3.7 isImageTIFSupported

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

5.9.3.8 isScanningAvailable

Syntax	<i>boolean isScanningAvailable(void);</i>
Description	Returns TRUE if the device has the capability to scan image data from media (the <i>readImageCapability</i> 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	Initialized after
transportCapability	Int	R	

Constructor	Parameter	Parameter-Type
JxfsDepTransportCapability	transportCapability	Int

Method	Return	May be used after
<i>getProperty</i>	<i>Property</i>	
isTransportAvailable	Boolean	
isShutterAvailable	Boolean	

Event	May occur after
none	

5.10.2 Properties

5.10.2.1 transportCapability (R)

Type	<i>Int</i>
Initial Value	0
Description	Specifies transport capability of the depository as a combination of the following flags : JXFS_DEP_TRANSPORT JXFS_DEP_SHUTTER

5.10.3 Methods

5.10.3.1 isTransportAvailable

Syntax	<i>boolean isTransportAvailable(void);</i>
Description	Returns TRUE if a deposit transport mechanism is available (the <i>transportCapability</i> property contains the value JXFS_DEP_TRANSPORT).
Parameter	None
Exceptions	None
Event	No additional events are generated.

5.10.3.2 isShutterAvailable

Syntax	<i>boolean isShutterAvailable(void);</i>
Description	Returns TRUE if a deposit transport shutter is available (the <i>transportCapability</i> property contains the value JXFS_DEP_SHUTTER).
Parameter	None
Exceptions	None
Event	No additional events are generated.

6 Status Event Classes

If a device status changes one of the following classes is returned via a *StatusEvent*. This *xxxStatus*-Class is passed with the *details* property of the *StatusEvent*.

Each *xxxStatus*-Class provides several methods to query the changed device status.

6.1 JxfsThresholdStatus

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

6.1.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

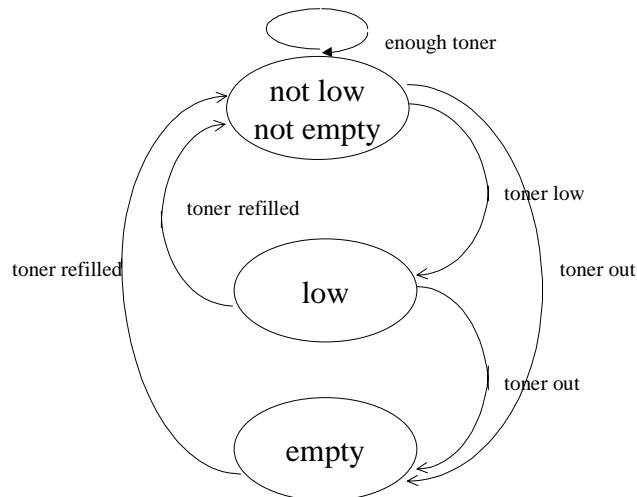
Property	Type	Access	Initialized after
thresholdState	int	R	

Constructor	Parameter	Parameter-Type
JxfsThresholdStatus	thresholdState	int

Method	Return	May be used after
isEmpty	<i>boolean</i>	
isFull	<i>boolean</i>	
isHigh	<i>boolean</i>	
isLow	<i>boolean</i>	
isUnknown	<i>boolean</i>	
toString	<i>String</i>	

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 :



6.2 JxfsDepRUnitStatus

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

6.2.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

Property	Type	Access	Initialized after
removableUnitStatus	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepRUnitStatus	removableUnitStatus	int

Method	Return	May be used after
isEmpty	<i>boolean</i>	
isFull	<i>boolean</i>	
isHigh	<i>boolean</i>	
isInOp	<i>boolean</i>	
isLow	<i>boolean</i>	
isMissing	<i>boolean</i>	
isNotSupported	<i>boolean</i>	
isOK	<i>boolean</i>	
isUnknown	<i>boolean</i>	
isUnlocked	<i>boolean</i>	

6.2.2 Properties

6.2.2.1 removableUnitStatus (R)

Type	<i>int</i>	
Initial Value	see Values below	
Description	Specifies the status of one of the removable units of the depository. Depending on device capability, <i>removableUnitStatus</i> 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 StatusEvent with a status value of:

Value	Meaning
JXFS_S_DEP_RUNIT	<i>removableUnitStatus</i> changed.

6.2.3 Methods

6.2.3.1 isEmpty

Syntax	<i>boolean isEmpty(void);</i>
Description	Returns TRUE if the removable supply unit is empty (the value of the <i>removableUnitStatus</i> property is JXFS_S_DEP_EMPTY).
Exceptions	None
Event	No additional events are generated.

6.2.3.2 isFull

Syntax	<i>boolean isFull(void);</i>
Description	Returns TRUE if the removable container unit is full (the value of the <i>removableUnitStatus</i> property is JXFS_S_DEP_FULL).
Exceptions	None
Event	No additional events are generated.

6.2.3.3 isHigh

Syntax	<i>boolean isHigh(void);</i>
Description	Returns TRUE if the removable container unit is almost full (the value of the <i>removableUnitStatus</i> property is JXFS_S_DEP_HIGH).
Exceptions	None
Event	No additional events are generated.

6.2.3.4 isInOp

Syntax	<i>boolean isInOp(void);</i>
Description	Returns TRUE if the removable container or supply unit is inoperable (the value of the <i>removableUnitStatus</i> property is JXFS_S_DEP_INOP).
Exceptions	None
Event	No additional events are generated.

6.2.3.5 isLow

Syntax	<i>boolean isLow(void);</i>
Description	Returns TRUE if the removable container unit is almost empty (the value of the <i>removableUnitStatus</i> property is JXFS_S_DEP_LOW).
Exceptions	None
Event	No additional events are generated.

6.2.3.6 isMissing

Syntax	<i>boolean isMissing(void);</i>
Description	Returns TRUE if the removable container or supply unit is missing (the value of the <i>removableUnitStatus</i> property is JXFS_S_DEP_MISSING).
Exceptions	None
Event	No additional events are generated.

6.2.3.7 isNotSupported

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

6.2.3.8 isOK

Syntax	<i>boolean isOK(void);</i>
---------------	----------------------------

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.

6.2.3.9 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.

6.2.3.10 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.

6.3 JxfsDepShutterStatus

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

6.3.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

Property	Type	Access	Initialized after
shutterStatus	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepShutterStatus	shutterStatus	int

Query	Return	May be used after
isShutterClosed	<i>boolean</i>	
isShutterJammed	<i>boolean</i>	
isShutterOpen	<i>boolean</i>	
isShutterNotSupported	<i>boolean</i>	
isShutterUnknown	<i>boolean</i>	

6.3.2 Properties

6.3.2.1 shutterStatus (R)

Type	<i>int</i>
Initial Value	see Values below
Description	Specifies the status of the transport shutter of the depository. Depending on device capability, <i>shutterStatus</i> 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 StatusEvent with a status value of:
Value	Meaning
JXFS_S_DEP_SHUTTER	<i>shutterStatus</i> changed.

6.3.3 Methods

6.3.3.1 isShutterClosed

Syntax	<i>boolean isShutterClosed(void);</i>
Description	Returns TRUE if the shutter is closed (the value of the <i>shutterStatus</i> property is JXFS_S_DEP_SHTCLOSED).
Exceptions	None
Event	No additional events are generated.

6.3.3.2 isShutterJammed

Syntax	<i>boolean isShutterJammed(void);</i>
Description	Returns TRUE if the shutter is jammed (the value of the <i>shutterStatus</i> property is JXFS_S_DEP_SHTJAMMED).
Exceptions	None
Event	No additional events are generated.

6.3.3.3 isShutterOpen

Syntax	<i>boolean isShutterOpen(void);</i>
Description	Returns TRUE if the shutter is closed (the value of the <i>shutterStatus</i> property is JXFS_S_DEP_SHTOPEN).
Exceptions	None
Event	No additional events are generated.

6.3.3.4 isNotSupported

Syntax	<i>boolean isNotSupported(void);</i>
Description	Returns TRUE if the physical device has no shutter (the value of the <i>shutterStatus</i> property is JXFS_S_DEP_NOTSUPPORTED).
Exceptions	None
Event	No additional events are generated.

6.3.3.5 isUnknown

Syntax	<i>boolean isUnknown(void);</i>
Description	Returns TRUE if the state of the shutter cannot be determined with the depository in its current state (the value of the <i>shutterStatus</i> property is JXFS_S_DEP_UNKNOWN).
Exceptions	None
Event	No additional events are generated.

6.4 JxfsDepUnitStatus

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

6.4.1 Summary

Implements : *Serializable*

Extends : *JxfsType*

Property	Type	Access	Initialized after
unitStatus	int	R	

Constructor	Parameter	Parameter-Type
JxfsDepUnitStatus	unitStatus	int

Method	Return	May be used after
isInOp	<i>boolean</i>	
isNotSupported	<i>boolean</i>	
isOK	<i>boolean</i>	
isUnknown	<i>boolean</i>	

6.4.2 Properties

6.4.2.1 unitStatus (R)

Type	<i>int</i>
Initial Value	see Values below
Description	Specifies the status of one of the fixed units of the depository. Depending on device capability, <i>unitStatus</i> 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 StatusEvent with a status value of:
Value	Meaning
JXFS_S_DEP_UNIT	<i>unitStatus</i> changed.

6.4.3 Methods

6.4.3.1 isInOp

Syntax	<i>boolean isInOp(void);</i>
Description	Returns TRUE if the unit is inoperable (the value of the <i>unitStatus</i> property is JXFS_S_DEP_INOP).
Exceptions	None
Event	No additional events are generated.

6.4.3.2 isOK

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

6.4.3.3 isNotSupported

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

6.4.3.4 isUnknown

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

6.5 JxfsDepStatus

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

6.5.1 Summary

Implements : *Serializable*

Extends : *JxfsStatus*

Property	Type	Access	Initialized after
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
<i>getProperty</i>	<i>Property</i>	

Event	May occur after
none	

6.5.2 Properties

6.5.2.1 containerStatus (R)

Type	<i>JxfsDepRUnitStatus</i>
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 StatusEvent with a status value of:
Value	Meaning
JXFS_S_DEP_CONTAINER	<i>containerStatus</i> changed.

6.5.2.2 envDispenserStatus (R)

Type	<i>JxfsDepUnitStatus</i>
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 StatusEvent with a status value of:
Value	Meaning
JXFS_S_DEP_ENVDISPENSER	<i>envDispenserStatus</i> changed.

6.5.2.3 envSupplyStatus (R)

Type
Description
Event

JxfsDepRUnitStatus

Specifies the state of the envelope supply unit.

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

Value	Meaning
JXFS_S_DEP_ENVSUPPLY	<i>envSupplyStatus</i> changed.

6.5.2.4 printerStatus (R)

Type
Description
Event

JxfsDepUnitStatus

Specifies the status of the printer.

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

Value	Meaning
JXFS_S_DEP_PRINTER	<i>printerStatus</i> changed.

6.5.2.5 scannerStatus (R)

Type
Description
Event

JxfsUnitStatus

Specifies the status of the scanner unit.

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

Value	Meaning
JXFS_S_DEP_SCANNER	<i>scannerStatus</i> changed.

6.5.2.6 shutterStatus (R)

Type
Description
Event

JxfsDepShutterStatus

Specifies the status of the transport shutter.

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

Value	Meaning
JXFS_S_DEP_SHUTTER	<i>shutterStatus</i> changed.

6.5.2.7 tonerStatus (R)

Type
Description
Event

JxfsThresholdStatus

Specifies the status of the toner supply.

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

Value	Meaning
JXFS_S_DEP_TONER	<i>tonerStatus</i> changed.

6.5.2.8 transportStatus (R)

Type
Description
Event

JxfsUnitStatus

Specifies the status of the deposit transport unit.

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

Value	Meaning
JXFS_S_DEP_TRANSPORT	<i>transportStatus</i> changed.

6.5.3 Events

6.5.3.1 IntermediateEvent

Interface	<i>jxfs.events.IntermediateListener</i>		
Method	intermediateOccurred(IntemmediateEvent <i>e</i>);		
Remarks	Issued to present intermediate information from the depository to the application		
Properties	Type	Name	Meaning
	int	code	The <i>code</i> 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 occurred 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.

6.5.3.2 StatusEvent

Interface	<i>jxfs.events.StatusListener</i>		
Method	statusOccurred(StatusEvent <i>e</i>);		
Remarks	Issued to present status data from the depository to the application		
Properties	Type	Name	Meaning
	int	status	The <i>status</i> value can be one of the following Codes:
Codes	Value		Meaning
	JXFS_S_DEVICE		Property <i>deviceStatus</i> changed.
	JXFS_S_DEP_CONTAINER		Property <i>containerStatus</i> changed.
	JXFS_S_DEP_ENVDISPENSER		Property <i>envDispenserStatus</i> changed
	JXFS_S_DEP_ENVSUPPLY		Property <i>envSupplyStatus</i> changed
	JXFS_S_DEP_PRINTER		Property <i>printerStatus</i> changed
	JXFS_S_DEP_SCANNER		Property <i>scannerStatus</i> changed
	JXFS_S_DEP_SHUTTER		Property <i>shutterStatus</i> changed
	JXFS_S_DEP_TONER		Property <i>tonerStatus</i> changed.
	JXFS_S_DEP_TRANSPORT		Property <i>transportStatus</i> changed

7 Codes

7.1 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.

7.2 Exception Codes

Value	Meaning
JXFS_E_CLOSED	Device has not been opened yet.
JXFS_E_PARAMETER_INVALID	An invalid parameter was given to the operation.
JXFS_E_NOT_SUPPORTED	Operation is not supported by device.
JXFS_E_REMOTE	Communication error during remote call.

7.3 Status Codes

7.3.1.1 General Status Codes

General Status Codes that specify a value change.

Value	Meaning
JXFS_S_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.

7.3.1.2 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.

7.3.1.3 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.

7.3.1.4 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 its current state.

7.3.1.5 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.

7.4 Constants

7.4.1.1 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.

7.4.1.2 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.

7.4.1.3 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.

7.4.1.4 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_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.

7.4.1.5 Retract Capability Codes

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

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

7.4.1.6 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_TRANSPORT	A deposit transport mechanism is available.
JXFS_DEP_SHUTTER	A deposit transport shutter is available.

7.5 Operation ID Codes

Following codes specify the operation which generated the OperationCompleteEvent.

Value	Method
JXFS_O_DEP_CLEAR_TRANSPORT	<i>clearTransport()</i>
JXFS_O_DEP_DISPENSE_ENVELOPE	<i>dispenseEnvelope()</i>
JXFS_O_DEP_ENTRY_ENVELOPE	<i>entryEnvelope()</i>
JXFS_O_DEP_READ_IMAGE	<i>readImage()</i>
JXFS_O_DEP_RETRACT_ENVELOPE	<i>retractEnvelope()</i>

8 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.

9 Index

actualNumOfDeposits	26
clearTransport	13
containerStatus	44
dispenseEnvelope	14
entryCapability	22
entryEnvelope	15
envDispenserStatus	44
envSupplyCapability	23
envSupplyStatus	45
getProperty	10
IJxfsDepositoryControl	11
imageData	25
imageType	25
IntermediateEvent	46
isBagDropSupported	22
isCodeLineSupported	32
isEmpty	38
isEnvelopeEntrySupported	22
isFull	38
isHigh	38
isImageBackSupported	32
isImageBMPSupported	32
isImageBothSupported	32
isImageFrontSupported	32
isImageMTFSupported	32
isImageTIFSupported	33
isInOp	38, 42
isLow	38
isManual	23
isMissing	38
isMotorized	23
isNone	24
isNotSupported	38, 41, 43
isOK	39, 43
isPrinterAvailable	27
isPrinterWithToner	28
isPrintOnRetractSupported	28
isRetractToDeposit	30
isScanningAvailable	33
isShutterAvailable	34
isShutterClosed	40
isShutterJammed	40
isShutterOpen	41
isTransportAvailable	34
isUnknown	39, 41, 43
isUnlocked	39
JXFS_DEP_BAGDROP	52
JXFS_DEP_ENVELOPE	52
JXFS_DEP_ENVMANUAL	52
JXFS_DEP_ENVMOTORIZED	52
JXFS_DEP_ENVNONE	52
JXFS_DEP_IMAGEBACK	53
JXFS_DEP_IMAGEBMP	53
JXFS_DEP_IMAGEBOTH	53
JXFS_DEP_IMAGEFRONT	53
JXFS_DEP_IMAGEMTF	53
JXFS_DEP_IMAGETIF	53
JXFS_DEP_PTRNONE	52
JXFS_DEP_PTRPRINTONRETRACT	52

JXFS_DEP_PTRRIBBON	52
JXFS_DEP_PTRTONER.....	52
JXFS_DEP_RETRACTTODEP	53
JXFS_DEP_RETRACTTOENV.....	53
JXFS_DEP_SHUTTER	53
JXFS_DEP_TRANSPORT	53
JXFS_E_CLOSED	49
JXFS_E_DEP_CODELINE	53
JXFS_E_DEP_CODELINEDATA.....	53
JXFS_E_DEP_CONT_MISSING.....	48
JXFS_E_DEP_DEPFULL.....	48
JXFS_E_DEP_DEPJAMMED.....	48
JXFS_E_DEP_DEPOSITERROR	46
JXFS_E_DEP_DEPUNKNOWN.....	48
JXFS_E_DEP_ENVEMPTY	48
JXFS_E_DEP_ENVJAMMED	48
JXFS_E_DEP_ENVSIZE	48
JXFS_E_DEP_NOCODELINE	48
JXFS_E_DEP_NOENV	48
JXFS_E_DEP_PTRFAIL.....	48
JXFS_E_DEP_SHTNOTCLOSED.....	48
JXFS_E_DEP_SHTNOTOPENED.....	48
JXFS_E_NOT_SUPPORTED.....	49
JXFS_E_PARAMETER_INVALID.....	49
JXFS_E_REMOTE.....	49
JXFS_I_DEP_ENVDEPOSITED	46
JXFS_I_DEP_ENVTAKEN	46
JXFS_I_DEP_MEDIA_INSERTED.....	46
JXFS_I_DEP_NO_MEDIA_PRESENT	46
JXFS_O_DEP_CLEAR_TRANSPORT	13, 54
JXFS_O_DEP_DISPENSE_ENVELOPE.....	14, 54
JXFS_O_DEP_ENTRY_ENVELOPE.....	15, 16, 54
JXFS_O_DEP_READ_IMAGE.....	17, 54
JXFS_O_DEP_RETRACT_ENVELOPE.....	19, 54
JXFS_S_BIN_EMPTY	50
JXFS_S_BIN_FULL.....	50
JXFS_S_BIN_HIGH.....	50
JXFS_S_BIN_LOW.....	50
JXFS_S_BIN_NOTSUPPORTED.....	50
JXFS_S_BIN_OK.....	50
JXFS_S_BIN_UNKNOWN	50
JXFS_S_DEP_CONTAINER	13, 16, 17, 19, 46, 50
JXFS_S_DEP_EMPTY.....	51
JXFS_S_DEP_ENVDISPENSER.....	14, 46, 50
JXFS_S_DEP_ENVSUPPLY	14, 46, 50
JXFS_S_DEP_FULL	51
JXFS_S_DEP_HIGH	51
JXFS_S_DEP_INOP.....	51
JXFS_S_DEP_LOW	51
JXFS_S_DEP_MISSING.....	51
JXFS_S_DEP_NOTSUPPORTED	51
JXFS_S_DEP_OK	51
JXFS_S_DEP_PRINTER.....	16, 19, 46, 50
JXFS_S_DEP_SCANNER.....	18, 46
JXFS_S_DEP_SHTCLOSED	51
JXFS_S_DEP_SHTJAMMED	51
JXFS_S_DEP_SHTOPEN	51
JXFS_S_DEP_SHUTTER	13, 14, 16, 18, 19, 46, 50
JXFS_S_DEP_TONER.....	16, 20, 46, 50
JXFS_S_DEP_TRANSPORT	13, 14, 16, 18, 20, 46, 50
JXFS_S_DEP_UNKNOWN	51
JXFS_S_DEP_UNLOCKED	51

JXFS_S_DEVICE	46, 50
JxfsDepEntryCapability	22
JxfsDepEnvSupplyCapability.....	23
JxfsDepImage.....	25
JxfsDepNumOfDeposits.....	26
JxfsDepPrintCapability	27
JxfsDepReadImageCapability	31
JxfsDepRetractCapability.....	30
JxfsDepRetractCount.....	29
JxfsDepRUnitStatus	37
JxfsDepShutterStatus.....	40
JxfsDepStatus	44
JxfsDepTransportCapability.....	34
JxfsDepUnitStatus	42
JxfsThresholdStatus.....	36
maxNumOfChars	27
OCDepReadImageEvent	20
printCapability.....	27
printerStatus	45
readImage	17
readImageCapability	31
removableUnitStatus	37
retractCapability	30
retractCount.....	29
retractEnvelope.....	19
scannerStatus.....	45
setProperty	10
shutterStatus	40, 45
status.....	12
Status Event Classes.....	36
StatusEvent.....	46
tonerStatus	45
transportCapability	34
transportStatus	45
unitStatus.....	42