



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

WORKSHOP AGREEMENT

CWA 13937-5

August 2000

ICS 35.240.40

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part
5: Cash Dispenser, Recycler and ATM 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 13937-5:2000 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/wkshp/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
3	CLASS DIAGRAM	7
4	CLASS AND INTERFACE SUMMARY	8
5	CLASS AND INTERFACE DETAILS	10
5.1.1	Access to properties.....	10
5.1.2	Exceptions	10
5.2	IJXFS CASHDISPENSERCONTROL	11
5.2.2	Properties.....	12
5.2.3	Methods	13
5.3	IJXFS CASHRECYCLERCONTROL.....	30
5.3.2	Methods	31
5.4	IJXFS ATMCONTROL.....	37
5.4.2	Properties.....	38
5.4.3	Methods	39
6	MDU - MINIMUM DISPENSE UNIT.....	44
7	SUPPORT CLASSES.....	45
7.2	JXFS CALIBRATEITEM	46
7.2.2	Properties.....	46
7.3	JXFS CAPABILITIES	47
7.3.2	Properties.....	49
7.4	JXFS CASHINORDER.....	53
7.4.2	Properties.....	53
7.5	JXFS CASHTYPE	54
7.5.2	Properties.....	54
7.6	JXFS CASHUNIT.....	55
7.6.2	Properties.....	55
7.6.3	Methods	55
7.7	JXFS CURRENCY.....	56
7.7.2	Properties.....	56
7.8	JXFS CURRENCYCODE	57
7.8.2	Properties.....	57
7.9	JXFS DELAY	58
7.9.2	Properties.....	58
7.10	JXFS DENOMINATION	59
7.10.2	Properties.....	59
7.10.3	Methods	60
7.11	JXFS DENOMINATIONITEM	61
7.11.2	Properties.....	61
7.12	JXFS DISPENSEORDER.....	62
7.12.2	Properties.....	62
7.13	JXFS DISPENSEREQUEST	64
7.13.2	Properties.....	64
7.14	JXFS LOGICALCASHUNIT	65
7.14.2	Properties.....	66
7.14.3	Methods	68
7.15	JXFS MIXENTRY.....	69
7.15.2	Properties.....	69
7.16	JXFS MIXINFO	70
7.16.2	Properties.....	70
7.17	JXFS MIXITEM.....	72
7.17.2	Properties.....	72

7.18	JXFSMIXTABLE	73
7.18.2	Properties.....	73
7.19	JXFSPHYSICALCASHUNIT.....	74
7.19.2	Properties.....	74
7.20	JXFSRETRACTAREA	76
7.20.2	Properties.....	76
7.21	JXFSTHRESHOLD	77
7.21.2	Properties.....	77
8	ERROR CODES.....	78
9	EXCEPTION CODES.....	79
10	STATUS EVENT CLASSES	80
10.1	JXFSCDRSTATUS	80
10.2	JXFSCASHTRAYSTATUS	80
10.3	JXFSCASHUNITSTATUS	80
10.4	JXFSDEVICESTATUS	81
10.5	JXFSDISPENSERSTATUS.....	81
10.6	JXFSDISPENSEORDERSTATUS.....	81
10.7	JXFSINTERMEDIATESTACKERSTATUS.....	81
10.8	JXFSPRESENTSTATUS	81
10.9	JXFSFAFEDOORSTATUS.....	81
10.10	JXFSHUTTERSTATUS.....	82
10.11	JXFSTHRESHOLDSTATUS	82
10.12	JXFSTRANSPORTSTATUS	82
10.13	JXFSVANDALISMSTATUS	82
11	STATUS CODES.....	83
12	INTERMEDIATE EVENT CODES	87
13	CONSTANTS.....	88
14	OPERATION ID CODES.....	91

1 Scope

This document describes the printer 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 Cash Dispenser, Recycler and ATM's the basic Device Control structure is extended with various properties and methods specific to this device which are described on the following pages.

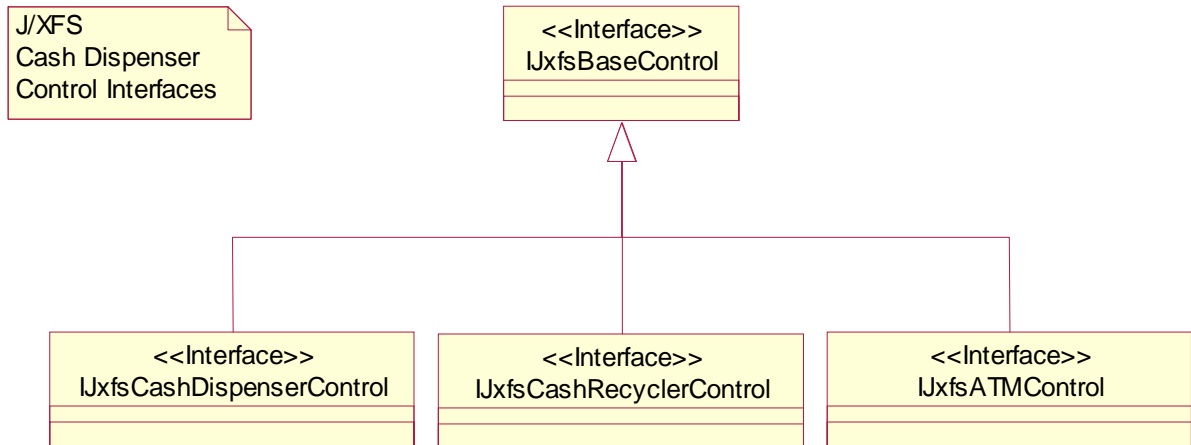
2 Overview

Cash Device Support within the J/XFS – API is available for the following device types:

- **Dispenser**
General dispense devices consist of components that allow the dispensing of cash, either bills or coins. This interface provides common functionality that is although used by the following device types.
- **Recycler**
A Recycler is primarily a Dispenser plus additional components that allow acceptance of cash as an input to the device.
- **ATM**
ATM's (Automated Teller Machine) inherit their functional behaviour from Dispenser and Recycler. They also have functions to support ATM-specific hardware.

3 Class Diagram

The following class diagram shows the overall layout of the Cash Dispenser, Recycler and ATM interfaces provided by J/XFS.



4 Class and Interface Summary

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

Class or Interface	Name	Description	Extends or Implements
Interface	IJxfBaseControl	Base interface for all device controls. Contains method declarations specific to all device controls.	
Interface	IJxfCashDispenserControl	Base interface for all cash dispenser controls. Contains method declarations specific to cash dispenser controls.	Extends: IJxfBaseControl
Interface	IJxfCashRecyclerControl	Base interface for all cash recycler controls. Contains method declarations specific to cash recycler controls.	Extends: IJxfBaseControl
Interface	IJxfATMControl	Base interface for all ATM controls. Contains method declarations specific to ATM controls.	Extends: IJxfBaseControl
Class	JxfCashDispenser	Class for cash dispenser control.	Implements: IJxfCashDispenserControl, IJxfBaseControl
Class	JxfCashRecycler	Class for cash recycler control.	Implements: IJxfCashDispenserControl, IJxfCashRecyclerControl, IJxfBaseControl
Class	JxfATM	Class for ATM control.	Implements: IJxfCashDispenserControl, IJxfCashRecyclerControl, IJxfATMControl, IJxfBaseControl

The following classes and interfaces are used by the J/XFS Cash Dispenser Device Services.

Class or Interface	Name	Description	Extends or Implements
Interface	IJxfBaseService	Base interface for all services.	
Interface	IJxfCashDispenserService	Base interface for all cash dispenser services. Contains method declarations specific to cash dispenser devices.	Extends: IJxfBaseService
Interface	IJxfCashRecyclerService	Base interface for all cash recycler services. Contains method declarations specific to cash recycler devices.	Extends: IJxfBaseService
Interface	IJxfATMService	Base interface for all ATM services. Contains method declarations specific to ATM devices.	Extends: IJxfBaseService

4.1.1.1 Remark on Device Services

The Device Service interface is common for all device services of a specific 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 controlID). This is always added as the last parameter in every operation.

5 Class and Interface Details

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

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

5.1.1 Access to properties

To read or write a property, an application must use the appropriate methods as defined in the JavaBeans specification.

During property access with a `setXXX`-method, an **`JxfsException`** containing an *errorCode* **`JXFS_E_INVALID_PARAMETER`** might be thrown. (Exception from this rule are properties of type **`boolean`**, which can never be invalid)

5.1.2 Exceptions

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

Exception	Value
<i>JxfsException</i>	<code>JXFS_E_CLAIMED</code>
	<code>JXFS_E_CLOSED</code>
	<code>JXFS_E_INVALID_PARAMETER</code>
	<code>JXFS_E_NOT_SUPPORTED</code>
	<code>JXFS_E_REMOTE</code>
	<code>JXFS_E_IO</code>
	<code>JXFS_E_ILLEGAL</code>
	<code>JXFS_E_FAILURE</code>
	<code>JXFS_E_TIMEOUT</code>

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

5.2 IJxfsCashDispenserControl

5.2.1.1 Summary

Extends	Implements
IJxfsBaseControl	

Property	Type	Access
capabilities	<i>JxfsCapabilities</i>	R
mixTable	<i>Vector of JxfsMixTable's</i>	RW
uvv	<i>boolean</i>	RW
currencies	<i>Vector of JxfsCurrency's</i>	R

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	
denominate	identificationID	
dispense	identificationID	
dispenseExec	identificationID	
startExchange	identificationID	
endExchange	identificationID	
openSafeDoor	identificationID	
calibrateCashUnit	identificationID	
getDateTime	identificationID	
setDateTime	identificationID	
queryOrder	identificationID	
reset	identificationID	
removeOrder	identificationID	
queryCashUnit	identificationID	
updateCashUnit	identificationID	

5.2.2 Properties

5.2.2.1 capabilities (R)

Type	<i>JxfsCapabilities</i>
Remarks	Used to keep complete information about all device Capabilities.

5.2.2.2 mixTables (RW)

Type	<i>Vector of JxfsMixTable's</i>
Remarks	Used to keep complete information about all MixTables.
Events	If the value of this property changes a <i>StatusEvent</i> is sent to all registered listeners with following data:
Field	Value
<i>status</i>	JXFS_S_CDR_MIXTABLE_CHANGED
<i>details</i>	-

5.2.2.3 uvv (RW)

Type	<i>boolean</i>
Remarks	UVV is a german abbreviation for „Unfallverhütungsvorschrift Kassen“. This is a regulation which describes the processing of dispensing cash according to german security rules. Defines the current mode for dispense operations. If set to true, delayed dispense (according to german security rules) is activated.

5.2.2.4 currencies (R)

Type	<i>Vector of JxfsCurrency's</i>
Remarks	Contains a vector of supported currencies.

5.2.3 Methods

5.2.3.1 denominate

Syntax	<i>identificationID denominate(int mixNumber, JxfsDenomination denomination, JxfsCurrency currency) throws JxfsException;</i>																																		
Remarks	<p>Denominates a specified amount of money. Cash can be retrieved from three different sources:</p> <ul style="list-style-type: none"> • cash dispenser • coin dispenser • teller's cash box <p>The configuration specifies which of these three sources are allowed to be used in the JxfsDenomination. For a Dispenser all three can be used. If the device used is an ATM, only the cash dispenser and, optionally, the coin dispenser can be available.</p>																																		
Parameter	Type	Name	Meaning																																
	<i>int</i>	mixNumber	ID of mixtable or algorithm to use																																
	<i>JxfsDenomination</i>	denomination	Specifies the amount to denominate.																																
	<i>JxfsCurrency</i>	currency	Specifies the Currency to use.																																
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>denominate</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_DENOMINATE</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>result:</i></td> <td>JXFS_RC_SUCCESSFUL</td> </tr> <tr> <td></td> <td>JXFS_E_TIMEOUT</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_RESET_REQUIRED</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_CASH_UNIT_ERROR</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_INVALID_CURRENCY</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_INVALID_DENOMINATION</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_INVALID_MIXNUMBER</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_NOT_DISPENSABLE</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_TOO_MANY_BILLS</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_TOO_MANY_COINS</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsDenomination</i> object</td> </tr> <tr> <td></td> <td>Specifies the calculated Denomination.</td> </tr> </tbody> </table>			Field	Value	<i>operationID</i>	JXFS_O_CDR_DENOMINATE	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL		JXFS_E_TIMEOUT		JXFS_E_CDR_RESET_REQUIRED		JXFS_E_CDR_CASH_UNIT_ERROR		JXFS_E_CDR_EXCHANGE_ACTIVE		JXFS_E_CDR_INVALID_CURRENCY		JXFS_E_CDR_INVALID_DENOMINATION		JXFS_E_CDR_INVALID_MIXNUMBER		JXFS_E_CDR_NOT_DISPENSABLE		JXFS_E_CDR_TOO_MANY_BILLS		JXFS_E_CDR_TOO_MANY_COINS	<i>data</i>	<i>JxfsDenomination</i> object		Specifies the calculated Denomination.
Field	Value																																		
<i>operationID</i>	JXFS_O_CDR_DENOMINATE																																		
<i>identificationID</i>	The corresponding ID																																		
<i>result:</i>	JXFS_RC_SUCCESSFUL																																		
	JXFS_E_TIMEOUT																																		
	JXFS_E_CDR_RESET_REQUIRED																																		
	JXFS_E_CDR_CASH_UNIT_ERROR																																		
	JXFS_E_CDR_EXCHANGE_ACTIVE																																		
	JXFS_E_CDR_INVALID_CURRENCY																																		
	JXFS_E_CDR_INVALID_DENOMINATION																																		
	JXFS_E_CDR_INVALID_MIXNUMBER																																		
	JXFS_E_CDR_NOT_DISPENSABLE																																		
	JXFS_E_CDR_TOO_MANY_BILLS																																		
	JXFS_E_CDR_TOO_MANY_COINS																																		
<i>data</i>	<i>JxfsDenomination</i> object																																		
	Specifies the calculated Denomination.																																		

5.2.3.2 dispense

Syntax	<i>identificationID dispense(JxfsDispenseRequest dispenseRequest) throws JxfsException;</i>										
Remarks	Dispenses the amount of money which is specified by the JxfsDenomination. The cash is dispensed at the side specified with the <i>position</i> property.										
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>JxfsDispenseRequest</i></td> <td>dispenseRequest</td> <td>Contains all parameter used for dispensing cash.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsDispenseRequest</i>	dispenseRequest	Contains all parameter used for dispensing cash.				
Type	Name	Meaning									
<i>JxfsDispenseRequest</i>	dispenseRequest	Contains all parameter used for dispensing cash.									
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>dispense</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_DISPENSE</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> </tbody> </table> <p><i>result:</i></p> <p>JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_DELAYED_DISPENSE JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_INVALID_MIXNUMBER JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_CMD_NOT_SUPPORTED</p> <p><i>data</i> JxfsDispenseOrder object Amongst other information, this carries a JxfsDenomination property. If a successful immediate dispense, or an error occurs, then this will return details of the actual cash dispensed. If the dispense is delayed (JXFS_E_CDR_DELAYED_DISPENSE result is returned by the event), then this will return details of the cash that will be dispensed following a successful call for the dispense order to the dispenseExec method. If the dispense is delayed, then the when property of the JxfsDispenseOrder will be set to the time from which the delay is started, and the delay property will give the total delay time in ms.</p> <p>StatusEvent When a <i>dispense</i> operation or a partial dispense completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> </tbody> </table> <p><i>details</i> JxfsCashUnitStatus object Updated cash units.</p> <p>Remark This status event is sent, if the dispense happend immediately and was not delayed.</p>	Field	Value	<i>operationID</i>	JXFS_O_CDR_DISPENSE	<i>identificationID</i>	The corresponding ID	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED
Field	Value										
<i>operationID</i>	JXFS_O_CDR_DISPENSE										
<i>identificationID</i>	The corresponding ID										
Field	Value										
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED										

StatusEvent

When a *dispense* operation or a partial dispense completed a *StatusEvent* can occur for all registered listeners with following data:

Field	Value
<i>status</i>	JXFS_S_CDR_DELAYED_DISPENSE
<i>details</i>	<i>JxfsDispenseOrderStatus</i> object Specifies among other data the time to delay in ms.
Remark	This status event is sent, if the dispense order is delayed for later dispense via <i>dispenseExec</i> .

StatusEvent

When a *dispense* operation or a partial dispense completed a *StatusEvent* can occur for all registered listeners with following data:

Field	Value
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD
<i>details</i>	<i>JxfsCashUnitStatus</i> object
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.

IntermediateEvent

Notifies the application, that the delay for an order has expired and that the order is now ready for dispense.

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_DELAYED_ORDER_READY
<i>data</i>	<i>JxfsDispenseOrderStatus</i> object Order is ready for dispense.

IntermediateEvent

When a *dispense* operation is divided into several sub-dispenses, an *IntermediateEvent* is sent to all registered listeners with following data:

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_PARTIAL_DISPENSE
<i>data</i>	<i>JxfsDispenseOrderStatus</i> object Contains information about dispensed amount of cash for each partial dispense.

IntermediateEvent

Notifies the application, that the dispensed cash is ready to be taken off the device.

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_CASH_AVAILABLE
<i>data</i>	none.

IntermediateEvent

Notifies the application, that the dispensed cash was taken off the device.

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_CASH_TAKEN
<i>data</i>	none.

5.2.3.3 dispenseExec

Syntax	<i>identificationID dispenseExec(JxfsDispenseOrder dispenseOrder) throws JxfsException;</i>																						
Remarks	Accepts an order, which should be ready for dispense.																						
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>JxfsDispenseOrder</i></td> <td>dispenseOrder</td> <td>Contains all parameter used for dispensing cash.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsDispenseOrder</i>	dispenseOrder	Contains all parameter used for dispensing cash.																
Type	Name	Meaning																					
<i>JxfsDispenseOrder</i>	dispenseOrder	Contains all parameter used for dispensing cash.																					
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>dispenseExec</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_DISPENSE_EXEC</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_ILLEGAL_DISPENSE_ORDER JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_INVALID_MIXNUMBER JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_UVV_IN_PROCESS JXFS_E_CDR_CMD_NOT_SUPPORTED</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsDispenseOrder</i> object Amongst other information, this carries a <i>JxfsDenomination</i> property. If a successful dispense, or an error occurs, then this will return details of the actual cash dispensed.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>dispenseExec</i> operation is completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>dispenseExec</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_ORDER_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsDispenseOrderStatus</i> object Dispensed Order.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_DISPENSE_EXEC	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_ILLEGAL_DISPENSE_ORDER JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_INVALID_MIXNUMBER JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_UVV_IN_PROCESS JXFS_E_CDR_CMD_NOT_SUPPORTED	<i>data</i>	<i>JxfsDispenseOrder</i> object Amongst other information, this carries a <i>JxfsDenomination</i> property. If a successful dispense, or an error occurs, then this will return details of the actual cash dispensed.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_ORDER_CHANGED	<i>details</i>	<i>JxfsDispenseOrderStatus</i> object Dispensed Order.
Field	Value																						
<i>operationID</i>	JXFS_O_CDR_DISPENSE_EXEC																						
<i>identificationID</i>	The corresponding ID																						
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_ILLEGAL_DISPENSE_ORDER JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_INVALID_MIXNUMBER JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_UVV_IN_PROCESS JXFS_E_CDR_CMD_NOT_SUPPORTED																						
<i>data</i>	<i>JxfsDispenseOrder</i> object Amongst other information, this carries a <i>JxfsDenomination</i> property. If a successful dispense, or an error occurs, then this will return details of the actual cash dispensed.																						
Field	Value																						
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																						
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																						
Field	Value																						
<i>status</i>	JXFS_S_CDR_ORDER_CHANGED																						
<i>details</i>	<i>JxfsDispenseOrderStatus</i> object Dispensed Order.																						

StatusEvent

When a *dispenseExec* operation or a partial dispense completed a *StatusEvent* can occur for all registered listeners with following data:

Field	Value
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD
<i>details</i>	<i>JxfsCashUnitStatus</i> object
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.

IntermediateEvent

When a *dispense* operation is divided into several sub-dispenses, an *IntermediateEvent* is sent to all registered listeners with following data:

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE_EXEC
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_PARTIAL_DISPENSE
<i>data</i>	<i>JxfsDispenseOrderStatus</i> object Contains information about dispensed amount of cash.

IntermediateEvent

Notifies the application, that the dispensed cash is ready to be taken off the device.

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE_EXEC
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_CASH_AVAILABLE
<i>data</i>	none.

IntermediateEvent

Notifies the application, that the dispensed cash was taken off the device.

Field	Value
<i>operationID</i>	JXFS_O_CDR_DISPENSE_EXEC
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_CASH_TAKEN
<i>data</i>	none.

5.2.3.4 startExchange

Syntax	<i>identificationID startExchange(Vector units) throws JxfsException;</i>		
Remarks	Used to start the exchange of cash units. No other method call than <i>endExchange</i> , <i>close</i> or <i>getAProperty</i> might be performed.		
Parameter	Type	Name	Meaning
	<i>Vector</i> of int's	units	Vector of int which specify the logical cash units to exchange.
Events	Additional events can be generated.		
	OperationCompleteEvent		
	When a <i>startExchange</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with following data:		
	Field	Value	
	<i>operationID</i>	JXFS_O_CDR_START_EXCHANGE	
	<i>identificationID</i>	The corresponding ID	
	<i>result:</i>	JXFS_RC_SUCCESSFUL	
		JXFS_E_CDR_CASH_DEVICE_ERROR	
		JXFS_E_CDR_RESET_REQUIRED	
		JXFS_E_CDR_CASH_UNIT_ERROR	
		JXFS_E_CDR_EXCHANGE_ACTIVE	
	<i>data</i>	<i>JxfsCashUnit</i> object	
		Current cash units.	

5.2.3.5 endExchange

Syntax	<i>identificationID endExchange(JxfsCashUnit cashUnit) throws JxfsException;</i>																								
Remarks	Set actual Cash Unit and put dispenser back into an operational state. It will now accept regular method calls.																								
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>JxfsCashUnit</i></td> <td>cashUnit</td> <td>Update information for the cash units.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsCashUnit</i>	cashUnit	Update information for the cash units.																		
Type	Name	Meaning																							
<i>JxfsCashUnit</i>	cashUnit	Update information for the cash units.																							
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>endExchange</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_END_EXCHANGE</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_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_NO_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsCashUnit</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>endExchange</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>endExchange</i> completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_END_EXCHANGE	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_NO_EXCHANGE_ACTIVE	<i>data</i>	<i>JxfsCashUnit</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.
Field	Value																								
<i>operationID</i>	JXFS_O_CDR_END_EXCHANGE																								
<i>identificationID</i>	The corresponding ID																								
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_NO_EXCHANGE_ACTIVE																								
<i>data</i>	<i>JxfsCashUnit</i> object Updated cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																								

5.2.3.6 openSafeDoor

Syntax	<i>identificationID openSafeDoor() throws JxfsException;</i>																																				
Remarks	This command controls the time lock for the safe door. It sends the currently configured value for the safe door timer to the device. This configuration parameter is vendor dependent.																																				
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>openSafeDoor</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_OPEN_SAFE_DOOR</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td></td> </tr> </tbody> </table> <p>StatusEvent When a <i>openSafeDoor</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>IntermediateEvent When a <i>openSafeDoor</i> operation is in progress an <i>IntermediateEvent</i> is sent to all registered listeners with 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_CDR_OPEN_SAFE_DOOR</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_I_CDR_SAFE_DOOR_LOCKED</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsDelay</i> object Specifies the delay of time until safe door can be opened in milliseconds.</td> </tr> </tbody> </table> <p>IntermediateEvent When a <i>openSafeDoor</i> operation is completed an <i>IntermediateEvent</i> is sent to all registered listeners with 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_CDR_OPEN_SAFE_DOOR</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_I_CDR_SAFE_DOOR_UNLOCKED</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsDelay</i> object Specifies the period of time while the safe door can be opened in milliseconds</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_OPEN_SAFE_DOOR	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>		Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>operationID</i>	JXFS_O_CDR_OPEN_SAFE_DOOR	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_CDR_SAFE_DOOR_LOCKED	<i>data</i>	<i>JxfsDelay</i> object Specifies the delay of time until safe door can be opened in milliseconds.	Field	Value	<i>operationID</i>	JXFS_O_CDR_OPEN_SAFE_DOOR	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_CDR_SAFE_DOOR_UNLOCKED	<i>data</i>	<i>JxfsDelay</i> object Specifies the period of time while the safe door can be opened in milliseconds
Field	Value																																				
<i>operationID</i>	JXFS_O_CDR_OPEN_SAFE_DOOR																																				
<i>identificationID</i>	The corresponding ID																																				
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE																																				
<i>data</i>																																					
Field	Value																																				
<i>status</i>	JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED																																				
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																																				
Field	Value																																				
<i>operationID</i>	JXFS_O_CDR_OPEN_SAFE_DOOR																																				
<i>identificationID</i>	The corresponding ID																																				
<i>reason</i>	JXFS_I_CDR_SAFE_DOOR_LOCKED																																				
<i>data</i>	<i>JxfsDelay</i> object Specifies the delay of time until safe door can be opened in milliseconds.																																				
Field	Value																																				
<i>operationID</i>	JXFS_O_CDR_OPEN_SAFE_DOOR																																				
<i>identificationID</i>	The corresponding ID																																				
<i>reason</i>	JXFS_I_CDR_SAFE_DOOR_UNLOCKED																																				
<i>data</i>	<i>JxfsDelay</i> object Specifies the period of time while the safe door can be opened in milliseconds																																				

5.2.3.7 calibrateCashUnit

Syntax	<i>identificationID calibrateJxfsCashUnit(JxfsCalibrateItem calibrateItem) throws JxfsException;</i>																
Remarks	This command is used to initialize the reference value of a cash unit. It will action a vendor dependent sequence of hardware events which will calibrate the physical cash unit. This is necessary if a new type of bank note is put into the cash unit. By this command the cash unit gets the new measures of the bank notes.																
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>JxfsCalibrateItem</i></td> <td>calibrateItem</td> <td>CalibrateItem to use.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsCalibrateItem</i>	calibrateItem	CalibrateItem to use.										
Type	Name	Meaning															
<i>JxfsCalibrateItem</i>	calibrateItem	CalibrateItem to use.															
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>calibrateJxfsCashUnit</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_CALIBRATE_CASH_UNIT</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td><i>java.util.Vector</i> object Updated JxfsCalibrateItems.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>calibrateJxfsCashUnit</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_CALIBRATE_CASH_UNIT	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>	<i>java.util.Vector</i> object Updated JxfsCalibrateItems.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.
Field	Value																
<i>operationID</i>	JXFS_O_CDR_CALIBRATE_CASH_UNIT																
<i>identificationID</i>	The corresponding ID																
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE																
<i>data</i>	<i>java.util.Vector</i> object Updated JxfsCalibrateItems.																
Field	Value																
<i>status</i>	JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED																
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																

5.2.3.8 getDateTime

Syntax
Remarks
Events

identificationID *getDateTime()* *throws JxfsException;*

Get device date and time.

Additional events can be generated.

OperationCompleteEvent

When a *getDateTime* operation is completed an

OperationCompleteEvent is sent to all registered listeners with following data:

Field	Value
<i>operationID</i>	JXFS_O_CDR_GET_DATE_TIME
<i>identificationID</i>	The corresponding ID
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CMD_NOT_SUPPORTED
<i>data</i>	<i>Date</i> object Current date and time of device.

5.2.3.9 setDateTime

Syntax	<i>identificationID setDateTime(Date date) throws JxfsException;</i>		
Remarks	Set device date and time. More and more devices were equipped with computer systems that have their own real time clock. The usage of this command is to synchronize this internal device clock with other systems.		
Parameter	Type	Name	Meaning
	<i>java.util.Date</i>	date	Date and time device is set to.
Events	Additional events can be generated.		
	OperationCompleteEvent		
	When a <i>setDateTime</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with following data:		
	Field	Value	
	<i>operationID</i>	JXFS_O_CDR_SET_DATE_TIME	
	<i>identificationID</i>	The corresponding ID	
	<i>result:</i>	JXFS_RC_SUCCESSFUL	
		JXFS_E_CDR_RESET_REQUIRED	
		JXFS_E_CDR_EXCHANGE_ACTIVE	
		JXFS_E_CDR_CMD_NOT_SUPPORTED	
	<i>data</i>	Date object	Previous date and time of device.
	StatusEvent		
	When a <i>setDateTime</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with following data:		
	Field	Value	
	<i>status</i>	JXFS_S_CDR_DATE_TIME_CHANGED	
	<i>details</i>	Date object	Updated device date and time.

5.2.3.10 queryCashUnit

Syntax
Remarks
Events

identificationID queryCashUnit() throws JxfsException;

Retrieve the current cash units.

Additional events can be generated.

OperationCompleteEvent

When a *queryCashUnit* operation is completed an

OperationCompleteEvent is sent to all registered listeners with following data:

Field	Value
<i>operationID</i>	JXFS_O_CDR_QUERY_CASHUNIT
<i>identificationID</i>	The corresponding ID
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CMD_NOT_SUPPORTED
<i>data</i>	<i>JxfsCashUnit</i> object Current cash units.

5.2.3.11 queryOrder

Syntax	<i>identificationID queryOrder(int orderType) throws JxsException;</i>																					
Remarks	This method is used to retrieve four different lists of dispense orders.																					
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>orderType</td> <td>specifies the list to retrieve.</td> </tr> <tr> <td>Value</td> <td></td> <td>Meaning</td> </tr> <tr> <td>JXFS_C_CDR_DO_DISPENSABLE</td> <td></td> <td>Orders ready for processing.</td> </tr> <tr> <td>JXFS_C_CDR_DO_DELAYED</td> <td></td> <td>All orders in delay queue.</td> </tr> <tr> <td>JXFS_C_CDR_DO_LAQ</td> <td></td> <td>All orders in Large Amount Queue.</td> </tr> <tr> <td>JXFS_C_CDR_DO_ALL</td> <td></td> <td>All orders in all queues.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>int</i>	orderType	specifies the list to retrieve.	Value		Meaning	JXFS_C_CDR_DO_DISPENSABLE		Orders ready for processing.	JXFS_C_CDR_DO_DELAYED		All orders in delay queue.	JXFS_C_CDR_DO_LAQ		All orders in Large Amount Queue.	JXFS_C_CDR_DO_ALL		All orders in all queues.
Type	Name	Meaning																				
<i>int</i>	orderType	specifies the list to retrieve.																				
Value		Meaning																				
JXFS_C_CDR_DO_DISPENSABLE		Orders ready for processing.																				
JXFS_C_CDR_DO_DELAYED		All orders in delay queue.																				
JXFS_C_CDR_DO_LAQ		All orders in Large Amount Queue.																				
JXFS_C_CDR_DO_ALL		All orders in all queues.																				
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>queryOrder</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_QUERY_ORDER</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_CDR_CASHDEVICE_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td>Vector of JxsDispenseOrder objects Vector of selected Orders.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_QUERY_ORDER	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHDEVICE_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>	Vector of JxsDispenseOrder objects Vector of selected Orders.											
Field	Value																					
<i>operationID</i>	JXFS_O_CDR_QUERY_ORDER																					
<i>identificationID</i>	The corresponding ID																					
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHDEVICE_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE																					
<i>data</i>	Vector of JxsDispenseOrder objects Vector of selected Orders.																					

5.2.3.12 reset

Syntax	<i>identificationID reset() throws JxfsException;</i>
Remarks	This method is used to reset the device and put it into a defined operational state.
Events	Additional events can be generated. OperationCompleteEvent When a <i>reset</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with following data:
Field	Value
<i>operationID</i>	JXFS_O_CDR_RESET
<i>identificationID</i>	The corresponding ID
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASH_DEVICE_ERROR JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE

5.2.3.13 removeOrder

Syntax	<i>identificationID removeOrder(JxfsDispenseOrder dispenseOrder) throws JxfsException;</i>																
Remarks	This method is used to remove a dispense order from the lists of dispense orders.																
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>JxfsDispenseOrder</i></td> <td>dispenseOrder</td> <td>specifies the dispenseOrder to remove from one of the queues: LAQ, Dispensable or Delayed.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsDispenseOrder</i>	dispenseOrder	specifies the dispenseOrder to remove from one of the queues: LAQ, Dispensable or Delayed.										
Type	Name	Meaning															
<i>JxfsDispenseOrder</i>	dispenseOrder	specifies the dispenseOrder to remove from one of the queues: LAQ, Dispensable or Delayed.															
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>removeOrder</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_REMOVE_ORDER</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_CDR_ILLEGAL_DISPENSE_ORDER JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsDispenseOrder</i> object Removed Order.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>removeOrder</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_ORDER_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsDispenseOrderStatus</i> object Contains removed Order.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_REMOVE_ORDER	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_ILLEGAL_DISPENSE_ORDER JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>	<i>JxfsDispenseOrder</i> object Removed Order.	Field	Value	<i>status</i>	JXFS_S_CDR_ORDER_CHANGED	<i>details</i>	<i>JxfsDispenseOrderStatus</i> object Contains removed Order.
Field	Value																
<i>operationID</i>	JXFS_O_CDR_REMOVE_ORDER																
<i>identificationID</i>	The corresponding ID																
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_ILLEGAL_DISPENSE_ORDER JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE																
<i>data</i>	<i>JxfsDispenseOrder</i> object Removed Order.																
Field	Value																
<i>status</i>	JXFS_S_CDR_ORDER_CHANGED																
<i>details</i>	<i>JxfsDispenseOrderStatus</i> object Contains removed Order.																

5.2.3.14 updateCashUnit

Syntax	<i>identificationID updateCashUnit(JxfsCashUnit cashUnit) throws JxfsException;</i>																								
Remarks	Replace current cash units.																								
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>JxfsCashUnit</i></td> <td>cashUnit</td> <td>Cash unit of device.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsCashUnit</i>	cashUnit	Cash unit of device.																		
Type	Name	Meaning																							
<i>JxfsCashUnit</i>	cashUnit	Cash unit of device.																							
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>updateCashUnit</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_UPDATE_CASHUNIT</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_CDR_CASH_UNIT_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CMD_NOT_SUPPORTED</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsCashUnit</i> object Current cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>updateCashUnit</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object All cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>updateCashUnit</i> operation completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_UPDATE_CASHUNIT	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CMD_NOT_SUPPORTED	<i>data</i>	<i>JxfsCashUnit</i> object Current cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object All cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.
Field	Value																								
<i>operationID</i>	JXFS_O_CDR_UPDATE_CASHUNIT																								
<i>identificationID</i>	The corresponding ID																								
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CMD_NOT_SUPPORTED																								
<i>data</i>	<i>JxfsCashUnit</i> object Current cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object All cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																								

5.3 IJxfsCashRecyclerControl

5.3.1.1 Summary

Extends	Implements
IJxfsBaseControl	

Method	Return	May use after
cashInStart	identificationID	
cashIn	identificationID	
cashInEnd	identificationID	
cashInRollback	identificationID	
empty	identificationID	

5.3.2 Methods

5.3.2.1 cashInStart

Syntax	<i>identificationID cashInStart(int position) throws JxfsException;</i>										
Remarks	Each cash in procedure has to be handled as a transaction that can be rolled back in any case if a difference occurs between the amount counted by the device and the amount the teller inserted. This command is used to start the cash in transaction at the defined input position.										
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>position</td> <td>Input position used during <i>cashIn</i>.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>int</i>	position	Input position used during <i>cashIn</i> .				
Type	Name	Meaning									
<i>int</i>	position	Input position used during <i>cashIn</i> .									
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>cashInStart</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_CASH_IN_START</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td>none</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_CASH_IN_START	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>	none
Field	Value										
<i>operationID</i>	JXFS_O_CDR_CASH_IN_START										
<i>identificationID</i>	The corresponding ID										
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_EXCHANGE_ACTIVE										
<i>data</i>	none										

5.3.2.2 cashIn

Syntax	<i>identificationID</i> cashIn (<i>JxfsCashInOrder</i> <i>order</i>) throws <i>JxfsException</i> ;																								
Remarks	Accept cash from the input slot.																								
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>JxfsCashInOrder</i></td> <td><i>order</i></td> <td>Specifies the notes or coins to accept.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsCashInOrder</i>	<i>order</i>	Specifies the notes or coins to accept.																		
Type	Name	Meaning																							
<i>JxfsCashInOrder</i>	<i>order</i>	Specifies the notes or coins to accept.																							
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>cashIn</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_CASH_IN</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_INVALID_BILL JXFS_E_CDR_INVALID_COIN JXFS_E_CDR_INPUT_REFUSED</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsCashInOrder</i> object Accepted cash.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>cashIn</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>cashIn</i> operation completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_CASH_IN	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_INVALID_BILL JXFS_E_CDR_INVALID_COIN JXFS_E_CDR_INPUT_REFUSED	<i>data</i>	<i>JxfsCashInOrder</i> object Accepted cash.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.
Field	Value																								
<i>operationID</i>	JXFS_O_CDR_CASH_IN																								
<i>identificationID</i>	The corresponding ID																								
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS JXFS_E_CDR_INVALID_BILL JXFS_E_CDR_INVALID_COIN JXFS_E_CDR_INPUT_REFUSED																								
<i>data</i>	<i>JxfsCashInOrder</i> object Accepted cash.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																								

5.3.2.3 cashInEnd

Syntax	<i>identificationID cashInEnd() throws JxfsException;</i>																								
Remarks	Each cash in procedure has to be handled as a transaction that can be rolled back if a difference occurs between the amount counted by the device and the amount the teller inserted. This command is used to end the cash in transaction.																								
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>cashInEnd</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_CASH_IN_END</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_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsCashInOrder</i> object Total amount and Denomination cashed in since <i>cashInStart</i>.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>cashInEnd</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>cashInEnd</i> operation completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_CASH_IN_END	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED	<i>data</i>	<i>JxfsCashInOrder</i> object Total amount and Denomination cashed in since <i>cashInStart</i> .	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.
Field	Value																								
<i>operationID</i>	JXFS_O_CDR_CASH_IN_END																								
<i>identificationID</i>	The corresponding ID																								
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED																								
<i>data</i>	<i>JxfsCashInOrder</i> object Total amount and Denomination cashed in since <i>cashInStart</i> .																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																								

5.3.2.4 cashInRollback

Syntax	<i>identificationID</i> cashInRollback () <i>throws JxfsException</i> ;																
Remarks	Each cash in procedure has to be handled as a transaction that can be rolled back if a difference occurs between the amount counted by the device and the amount the teller inserted. This command is used to roll back the cash in transaction. All the notes cashed in since the last <i>cashInStart</i> command are returned to the teller. If an device does not have this capability, it returns 0 number of bills in the Denomination structure.																
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>cashInRollback</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_CASH_IN_ROLLBACK</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsCashInOrder</i> object This represents the amount of cash that is returned by this action.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>cashInRollback</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_CASH_IN_ROLLBACK	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS	<i>data</i>	<i>JxfsCashInOrder</i> object This represents the amount of cash that is returned by this action.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.
Field	Value																
<i>operationID</i>	JXFS_O_CDR_CASH_IN_ROLLBACK																
<i>identificationID</i>	The corresponding ID																
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NO_CASHIN_STARTED JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS																
<i>data</i>	<i>JxfsCashInOrder</i> object This represents the amount of cash that is returned by this action.																
Field	Value																
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																

5.3.2.5 empty

Syntax	<i>identificationID empty(JxfsDispenseRequest dispenseRequest) throws JxfsException;</i>																								
Remarks	This method is used to empty the cash dispenser of a particular Denomination of bills.																								
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>JxfsDispenseRequest</i></td> <td>dispenseRequest</td> <td>Contains all parameter used to empty the device.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsDispenseRequest</i>	dispenseRequest	Contains all parameter used to empty the device.																		
Type	Name	Meaning																							
<i>JxfsDispenseRequest</i>	dispenseRequest	Contains all parameter used to empty the device.																							
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>empty</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_EMPTY</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_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_DELAYED_DISPENSE JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS</td> </tr> <tr> <td><i>data</i></td> <td><i>JxfsDispenseOrder</i> object Dispensed cash.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>empty</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> </tbody> </table> <p>StatusEvent When a <i>empty</i> operation completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_EMPTY	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_DELAYED_DISPENSE JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS	<i>data</i>	<i>JxfsDispenseOrder</i> object Dispensed cash.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.
Field	Value																								
<i>operationID</i>	JXFS_O_CDR_EMPTY																								
<i>identificationID</i>	The corresponding ID																								
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_CASH_UNIT_ERROR JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_DELAYED_DISPENSE JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_INVALID_CURRENCY JXFS_E_CDR_INVALID_DENOMINATION JXFS_E_CDR_NOT_DISPENSABLE JXFS_E_CDR_TOO_MANY_BILLS JXFS_E_CDR_TOO_MANY_COINS																								
<i>data</i>	<i>JxfsDispenseOrder</i> object Dispensed cash.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																								

StatusEvent

When a *empty* operation or a partial dispense completed a *StatusEvent* can occur for all registered listeners with following data:

Field	Value
<i>status</i>	JXFS_S_CDR_DELAYED_DISPENSE
<i>details</i>	<i>JxfsDispenseOrderStatus</i> object Specifies among other data the time to delay in ms.
Remark	This status event is sent, if the dispense order is delayed for later dispense via <i>dispenseExec</i> or <i>empty</i> .

IntermediateEvent

Notifies the application, that the delay for an order has expired and that the order is now ready for dispense.

Field	Value
<i>operationID</i>	JXFS_O_CDR_EMPTY
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_DELAYED_ORDER_READY
<i>data</i>	<i>JxfsDispenseOrderStatus</i> object Order is ready for dispense.

IntermediateEvent

When a *dispense* operation is divided into several sub-dispenses, an *IntermediateEvent* is sent to all registered listeners with following data:

Field	Value
<i>operationID</i>	JXFS_O_CDR_EMPTY
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_PARTIAL_DISPENSE
<i>data</i>	<i>JxfsDispenseOrderStatus</i> object Contains information about dispensed amount of cash.

IntermediateEvent

Notifies the application, that the dispensed cash was taken off the device.

Field	Value
<i>operationID</i>	JXFS_O_CDR_EMPTY
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_CASH_TAKEN
<i>data</i>	none.

5.4 JxfsATMControl

5.4.1.1 Summary

Extends	Implements
IJxfsBaseControl	

Property	Type	Access
retractArea	<i>JxfsRetractArea</i>	R

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	
present	identificationID	
reject	identificationID	
retract	identificationID	
shutterOpen	identificationID	

5.4.2 Properties

5.4.2.1 retractArea (R)

Type
Remarks
Events

JxfsRetractArea

If the value of this property changes a *StatusEvent* is sent to all registered listeners with following data:

Field	Value
<i>status</i>	JXFS_S_CDR_RETRACT_AREA_CHANGED
<i>details</i>	-

5.4.3 Methods

5.4.3.1 present

Syntax	<i>identificationID present() throws JxfsException;</i>																																						
Remarks	<p>This command causes presentation of the cash. It can be used only following the <i>dispense</i> method.</p> <p>The command completes when the bills are positioned at the exit slot of the device. A status event is generated to report the user has removed the bills. If no event is received within a reasonable time period, the application should send a <i>retract</i> method to clear the bills from the exit. On devices which do not have the ability to detect when bills are taken the service event is generated as soon as the bills are available to the user.</p>																																						
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>present</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_PRESENT</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>result:</i></td> <td>JXFS_RC_SUCCESSFUL</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_RESET_REQUIRED</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_CASH_UNIT_ERROR</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_CASHIN_ACTIVE</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td></td> <td>JXFS_E_CDR_NO_BILLS</td> </tr> </tbody> </table> <p><i>data</i></p> <p>IntermediateEvent Notifies the application, that the dispensed cash is ready to be taken off the device.</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_CDR_PRESENT</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_I_CDR_CASH_AVAILABLE</td> </tr> <tr> <td><i>data</i></td> <td>none.</td> </tr> </tbody> </table> <p>IntermediateEvent Notifies the application, that the dispensed cash was taken off the device.</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_CDR_PRESENT</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_I_CDR_CASH_TAKEN</td> </tr> <tr> <td><i>data</i></td> <td>none.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_PRESENT	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL		JXFS_E_CDR_RESET_REQUIRED		JXFS_E_CDR_CASH_UNIT_ERROR		JXFS_E_CDR_CASHIN_ACTIVE		JXFS_E_CDR_EXCHANGE_ACTIVE		JXFS_E_CDR_NO_BILLS	Field	Value	<i>operationID</i>	JXFS_O_CDR_PRESENT	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_CDR_CASH_AVAILABLE	<i>data</i>	none.	Field	Value	<i>operationID</i>	JXFS_O_CDR_PRESENT	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_CDR_CASH_TAKEN	<i>data</i>	none.
Field	Value																																						
<i>operationID</i>	JXFS_O_CDR_PRESENT																																						
<i>identificationID</i>	The corresponding ID																																						
<i>result:</i>	JXFS_RC_SUCCESSFUL																																						
	JXFS_E_CDR_RESET_REQUIRED																																						
	JXFS_E_CDR_CASH_UNIT_ERROR																																						
	JXFS_E_CDR_CASHIN_ACTIVE																																						
	JXFS_E_CDR_EXCHANGE_ACTIVE																																						
	JXFS_E_CDR_NO_BILLS																																						
Field	Value																																						
<i>operationID</i>	JXFS_O_CDR_PRESENT																																						
<i>identificationID</i>	The corresponding ID																																						
<i>reason</i>	JXFS_I_CDR_CASH_AVAILABLE																																						
<i>data</i>	none.																																						
Field	Value																																						
<i>operationID</i>	JXFS_O_CDR_PRESENT																																						
<i>identificationID</i>	The corresponding ID																																						
<i>reason</i>	JXFS_I_CDR_CASH_TAKEN																																						
<i>data</i>	none.																																						

5.4.3.2 reject

Syntax	<i>identificationID reject(boolean present) throws JxfsException;</i>																																		
Remarks	Specifies if the rejected cash should be presented to the user at the position specified by the preceding <i>dispense</i> , <i>dispenseExec</i> or <i>calibrateCashUnit</i> method (present = true) or, whether the cash should be moved to the reject bin.																																		
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>boolean</i></td> <td>present</td> <td>Specifies if the cash should be presented to user using the specified position (=true) or, if the money should only be transported to the stacker (=false).</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>boolean</i>	present	Specifies if the cash should be presented to user using the specified position (=true) or, if the money should only be transported to the stacker (=false).																												
Type	Name	Meaning																																	
<i>boolean</i>	present	Specifies if the cash should be presented to user using the specified position (=true) or, if the money should only be transported to the stacker (=false).																																	
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>reject</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_REJECT</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_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td></td> </tr> </tbody> </table> <p>StatusEvent When a <i>reject</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>reject</i> operation completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table> <p>IntermediateEvent Notifies the application, that the dispensed cash is ready to be taken off the device.</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_CDR_REJECT</td> </tr> <tr> <td><i>identificationID</i></td> <td>The corresponding ID</td> </tr> <tr> <td><i>reason</i></td> <td>JXFS_I_CDR_CASH_AVAILABLE</td> </tr> <tr> <td><i>data</i></td> <td>none.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_REJECT	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>		Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.	Field	Value	<i>operationID</i>	JXFS_O_CDR_REJECT	<i>identificationID</i>	The corresponding ID	<i>reason</i>	JXFS_I_CDR_CASH_AVAILABLE	<i>data</i>	none.
Field	Value																																		
<i>operationID</i>	JXFS_O_CDR_REJECT																																		
<i>identificationID</i>	The corresponding ID																																		
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE																																		
<i>data</i>																																			
Field	Value																																		
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																																		
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																																		
Field	Value																																		
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																																		
<i>details</i>	<i>JxfsCashUnitStatus</i> object																																		
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																																		
Field	Value																																		
<i>operationID</i>	JXFS_O_CDR_REJECT																																		
<i>identificationID</i>	The corresponding ID																																		
<i>reason</i>	JXFS_I_CDR_CASH_AVAILABLE																																		
<i>data</i>	none.																																		

IntermediateEvent

Notifies the application, that the dispensed cash was taken off the device.

Field	Value
<i>operationID</i>	JXFS_O_CDR_REJECT
<i>identificationID</i>	The corresponding ID
<i>reason</i>	JXFS_I_CDR_CASH_TAKEN
<i>data</i>	none.

5.4.3.3 retract

Syntax	<i>identificationID retract(JxfsRetractArea retractArea) throws JxfsException;</i>																								
Remarks	This command allows the application to force cash that has been presented to be retracted. Not all cash dispensers support this capability. This method may only be called following a <i>dispense</i> or <i>dispenseExec</i> method, and the retract will be performed on the position specified by that preceding method.																								
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>JxfsRetractArea</i></td> <td>retractArea</td> <td>A vendor dependant parameter, specifying the retract area to which the notes will be withdrawn.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>JxfsRetractArea</i>	retractArea	A vendor dependant parameter, specifying the retract area to which the notes will be withdrawn.																		
Type	Name	Meaning																							
<i>JxfsRetractArea</i>	retractArea	A vendor dependant parameter, specifying the retract area to which the notes will be withdrawn.																							
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>retract</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_RETRACT</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_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE</td> </tr> <tr> <td><i>data</i></td> <td></td> </tr> </tbody> </table> <p>StatusEvent When a <i>retract</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object Updated cash units.</td> </tr> </tbody> </table> <p>StatusEvent When a <i>retract</i> operation completed a <i>StatusEvent</i> can occur for all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_CASHUNIT_THRESHOLD</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfsCashUnitStatus</i> object</td> </tr> <tr> <td>Remark</td> <td>This status event is sent, if a threshold change occurred for one or more cassettes.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_RETRACT	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE	<i>data</i>		Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED	<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.	Field	Value	<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD	<i>details</i>	<i>JxfsCashUnitStatus</i> object	Remark	This status event is sent, if a threshold change occurred for one or more cassettes.
Field	Value																								
<i>operationID</i>	JXFS_O_CDR_RETRACT																								
<i>identificationID</i>	The corresponding ID																								
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE																								
<i>data</i>																									
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_CHANGED																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object Updated cash units.																								
Field	Value																								
<i>status</i>	JXFS_S_CDR_CASHUNIT_THRESHOLD																								
<i>details</i>	<i>JxfsCashUnitStatus</i> object																								
Remark	This status event is sent, if a threshold change occurred for one or more cassettes.																								

5.4.3.4 shutterMove

Syntax	<i>identificationID shutterMove(boolean open, int position) throws JxfException;</i>																									
Remarks	This method allows the calling application to open and close the dispense shutter. The open parameter specifies in which direction the shutter should be moved. The position parameter determines for which dispense position the shutter is moved.																									
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>boolean</i></td> <td>open</td> <td>true – open shutter false – close shutter</td> </tr> <tr> <td><i>int</i></td> <td>position</td> <td>Specifies the output position to which side to dispense.</td> </tr> </tbody> </table> <table border="0"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Meaning</th> </tr> </thead> <tbody> <tr> <td>JXFS_C_CDR_POS_DEFAULT</td> <td>Use configured position</td> </tr> <tr> <td>JXFS_C_CDR_POS_LEFT</td> <td>Dispense to left side</td> </tr> <tr> <td>JXFS_C_CDR_POS_CENTER</td> <td>Dispense to the middle</td> </tr> <tr> <td>JXFS_C_CDR_POS_RIGHT</td> <td>Dispense to right side</td> </tr> <tr> <td>JXFS_C_CDR_POS_REJECT</td> <td>Dispense to reject container.</td> </tr> <tr> <td>JXFS_C_CDR_POS_TOP</td> <td>Used for shutter.</td> </tr> <tr> <td>JXFS_C_CDR_POS_BOTTOM</td> <td>Used for shutter.</td> </tr> </tbody> </table>	Type	Name	Meaning	<i>boolean</i>	open	true – open shutter false – close shutter	<i>int</i>	position	Specifies the output position to which side to dispense.	Value	Meaning	JXFS_C_CDR_POS_DEFAULT	Use configured position	JXFS_C_CDR_POS_LEFT	Dispense to left side	JXFS_C_CDR_POS_CENTER	Dispense to the middle	JXFS_C_CDR_POS_RIGHT	Dispense to right side	JXFS_C_CDR_POS_REJECT	Dispense to reject container.	JXFS_C_CDR_POS_TOP	Used for shutter.	JXFS_C_CDR_POS_BOTTOM	Used for shutter.
Type	Name	Meaning																								
<i>boolean</i>	open	true – open shutter false – close shutter																								
<i>int</i>	position	Specifies the output position to which side to dispense.																								
Value	Meaning																									
JXFS_C_CDR_POS_DEFAULT	Use configured position																									
JXFS_C_CDR_POS_LEFT	Dispense to left side																									
JXFS_C_CDR_POS_CENTER	Dispense to the middle																									
JXFS_C_CDR_POS_RIGHT	Dispense to right side																									
JXFS_C_CDR_POS_REJECT	Dispense to reject container.																									
JXFS_C_CDR_POS_TOP	Used for shutter.																									
JXFS_C_CDR_POS_BOTTOM	Used for shutter.																									
Events	<p>Additional events can be generated.</p> <p>OperationCompleteEvent When a <i>shutterMove</i> operation is completed an <i>OperationCompleteEvent</i> is sent to all registered listeners with 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_CDR_SHUTTER_MOVE</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_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_UNABLE_OPEN_SHUTTER</td> </tr> </tbody> </table> <p><i>data</i></p> <p>StatusEvent When a <i>shutterMove</i> operation is completed an <i>StatusEvent</i> is sent to all registered listeners with 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>status</i></td> <td>JXFS_S_CDR_SHUTTER_CHANGED</td> </tr> <tr> <td><i>details</i></td> <td><i>JxfShutterStatus</i> object New shutter status.</td> </tr> </tbody> </table>	Field	Value	<i>operationID</i>	JXFS_O_CDR_SHUTTER_MOVE	<i>identificationID</i>	The corresponding ID	<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_UNABLE_OPEN_SHUTTER	Field	Value	<i>status</i>	JXFS_S_CDR_SHUTTER_CHANGED	<i>details</i>	<i>JxfShutterStatus</i> object New shutter status.											
Field	Value																									
<i>operationID</i>	JXFS_O_CDR_SHUTTER_MOVE																									
<i>identificationID</i>	The corresponding ID																									
<i>result:</i>	JXFS_RC_SUCCESSFUL JXFS_E_CDR_CASHIN_ACTIVE JXFS_E_CDR_RESET_REQUIRED JXFS_E_CDR_EXCHANGE_ACTIVE JXFS_E_CDR_UNABLE_OPEN_SHUTTER																									
Field	Value																									
<i>status</i>	JXFS_S_CDR_SHUTTER_CHANGED																									
<i>details</i>	<i>JxfShutterStatus</i> object New shutter status.																									

6 MDU - Minimum Dispense Unit

Each monetary amount is expressed in terms of multiples of „Minimum Dispense Units“ (MDU).

6.1.1.1 Definitions

<i>Abbreviation</i>	<i>Description</i>
MDU	Minimum Dispense Unit
CU	Currency Unit, defined in ISO 4217
CE	Currency Exponent
MAP	Money Amount Parameter. Amount of cash expressed in MDUs.

<i>Currency Unit (CU) for ...</i>	<i>Country Code</i>	<i>Description</i>
German money	DEM	1 German Mark
Italian money	LIT	1 Italian Lira

<i>Currency Exponent (CE) for ...</i>	<i>Description</i>	<i>MDU equals</i>
German money	- 2	1 Pfennig
Italian money	+ 2	100 Lire

A MDU is equal to CU *times* CE.

A MAP relates to the amount of cash like: *Amount of cash* = $MAP * 10 ^ CE$.

6.1.1.2 Examples

Germany:

Country code	DEM
CU	1 German Mark (= 100 Pfennig)
CE	-2
MAP	10050

Amount of cash	$MAP * 10 ^ CE$
100,50 DEM	$10050 * 10 ^{-2}$

Italy:

Country code	LIT
CU	1 Italian Lira
CE	+2
MAP	150

Amount of cash	$MAP * 10 ^ CE$
15000 LIT	$150 * 10 ^ 2$

7 Support Classes

7.1.1.1 Summary

Class	Description
JxfsCalibrateItem	Data used by initialization and calibration of cash units.
JxfsCapabilities	Contains the Capabilities of a cash dispenser.
JxfsCashInOrder	This class specifies all data required for cashIn operations.
JxfsCashType	Used to differentiate between bills and coins.
JxfsCashUnit	Information about the status and contents of the logical and physical cash units.
JxfsCurrency	Defines a Currency.
JxfsCurrencyCode	Contains a 3-character string detailing a currency code as defined by the ISO standard.
JxfsDelay	Used for OpenSafeDoor operation.
JxfsDenomination	The JxfsDenomination holds a collection of JxfsDenominationItems that sum up to an amount of cash.
JxfsDenominationItem	A JxfsDenominationItem specifies a logical cash unit and the number of bills or coins that were dispensed from this unit or that should be deposited into this unit.
JxfsDispenseOrder	This class specifies all data required to perform a dispense operation.
JxfsDispenseRequest	This class specifies all data required for requesting a dispense operation.
JxfsLogicalCashUnit	Logical information about a cash unit.
JxfsMixEntry	Contains a reference to the logical cash unit and the number of bills/coins used in mixing.
JxfsMixInfo	Type for identifying mix algorithm and/or house mix tables.
JxfsMixItem	Specifies an amount used in a JxfsMixTable. The amount is expressed in MDU's.
JxfsMixTable	Contains complete description of one house mix table.
JxfsPhysicalCashUnit	Information about a physical cash unit.
JxfsRetractArea	TBD
JxfsThreshold	Defines cassette thresholds.

7.2 JxfsCalibrateItem

7.2.1.1 Usage

Data used by initialization and calibration of cash units. The vendor supplied service control is responsible for mapping from logical to physical cash units.

7.2.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
logicalNumber	<i>int</i>	RW
billsCount	<i>int</i>	RW
position	<i>int</i>	RW

Constructor	Parameter	Parameter-Type
JxfsCalibrateItem	<i>logicalNumber</i>	int
	<i>billsCount</i>	int
	<i>position</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.2.2 Properties

7.2.2.1 logicalNumber (RW)

Type	<i>int</i>
Remarks	This value specifies the number of the logical cash unit to be used during the initialization.

7.2.2.2 billsCount (RW)

Type	<i>int</i>
Remarks	On input this value specifies the number of bills to dispense.

7.2.2.3 position (RW)

Type	<i>int</i>
Remarks	Specifies the output position to dispense the note. (Defined as <i>dispense position codes</i>)

7.3 JxfsCapabilities

7.3.1.1 Usage

Used to query the JxfsCapabilities of a cash dispenser / recycler.

7.3.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
autoPresent	<i>boolean</i>	R
cdType	<i>int</i>	R
maxInBills	<i>int</i>	R
maxInCoins	<i>int</i>	R
maxOutBills	<i>int</i>	R
maxOutCoins	<i>int</i>	R
compound	<i>boolean</i>	R
shutterCmd	<i>boolean</i>	R
retract	<i>boolean</i>	R
safeDoorCmd	<i>boolean</i>	R
coins	<i>boolean</i>	R
cylinders	<i>boolean</i>	R
cashBox	<i>boolean</i>	R
refill	<i>boolean</i>	R
dispense	<i>boolean</i>	R
deposit	<i>boolean</i>	R
checkVandalism	<i>boolean</i>	R
intermediateStacker	<i>boolean</i>	R
billsTakenSensor	<i>boolean</i>	R
inputPositions	<i>int</i>	R
outputPositions	<i>int</i>	R
defaultInputPosition	<i>int</i>	R
defaultOutputPosition	<i>int</i>	R
silentAlarm	<i>boolean</i>	R
escrow	<i>boolean</i>	R
escrowSize	<i>int</i>	R
detector	<i>boolean</i>	R
baitTrap	<i>boolean</i>	R
vendorData	<i>java.lang.String</i>	R

Constructor	Parameter	Parameter-Type
JxfsCapabilities	<i>autoPresent</i>	boolean
	<i>cdType</i>	int
	<i>maxInBills</i>	int
	<i>maxInCoins</i>	int
	<i>maxOutBills</i>	int
	<i>maxOutCoins</i>	int
	<i>compound</i>	boolean
	<i>shutterCmd</i>	boolean
	<i>retract</i>	boolean
	<i>safeDoorCmd</i>	boolean
	<i>coins</i>	boolean
	<i>cylinders</i>	boolean
	<i>cashBox</i>	boolean
	<i>refill</i>	boolean
	<i>dispense</i>	boolean
	<i>deposit</i>	boolean
	<i>checkVandalism</i>	boolean
	<i>intermediateStacker</i>	boolean
	<i>billsTakenSensor</i>	boolean
	<i>inputPositions</i>	int
	<i>outputPositions</i>	int
	<i>defaultInputPosition</i>	int
	<i>defaultOutputPosition</i>	int
	<i>silentAlarm</i>	boolean
	<i>escrow</i>	boolean
	<i>escrowSize</i>	int
	<i>detector</i>	boolean
	<i>baitTrap</i>	boolean
	<i>vendorData</i>	java.lang.String

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>isProperty</i>	<i>Property</i>	

7.3.2 Properties

7.3.2.1 autoPresent (R)

Type	<i>boolean</i>
Remarks	This specifies whether cash will be automatically presented to the user on execution of a dispense (autoPresent set to true), or whether the cash will only be transported to the stacker. In the latter case, a present command will need to be issued following the dispense (or following each part of a multi-partition dispense). If this property is set to true, then the shutterCmd capability will be false, as it would not be possible for the calling application to determine when it should open the dispense shutter, due to the possibility for a dispense to be delayed.

7.3.2.2 cdType (R)

Type	<i>int</i>
Remarks	Type of device. One of the following values: JXFS_C_CDR_TYPE_NONE JXFS_C_CDR_TYPE_DISPENSER JXFS_C_CDR_TYPE_RECYCLER JXFS_C_CDR_TYPE_ATM

7.3.2.3 maxInBills (R)

Type	<i>int</i>
Remarks	Maximum number of bills to be accepted by one command.

7.3.2.4 maxInCoins (R)

Type	<i>int</i>
Remarks	Maximum number of coins to be accepted by one command.

7.3.2.5 maxOutBills (R)

Type	<i>int</i>
Remarks	Maximum number of bills to be dispensed by one command.

7.3.2.6 maxOutCoins (R)

Type	<i>int</i>
Remarks	Maximum number of coins to be dispensed by one command.

7.3.2.7 compound (R)

Type	<i>boolean</i>
Remarks	Is logical device part of compound physical device.

7.3.2.8 shutterCmd (R)

Type	<i>boolean</i>
Remarks	Defines, if the shutter be accessed by commands. If this property is set to true, then the autoPresent capability will be false, as it would not be possible for the calling application to determine when it should open the dispense shutter, due to the possibility for a dispense to be delayed.

7.3.2.9 retract (R)

Type	<i>boolean</i>
Remarks	Can the cash dispenser retract presented bills.

7.3.2.10 safeDoorCmd (R)

Type	<i>boolean</i>
Remarks	Has safe door electrical time lock command.

7.3.2.11 coins (R)

Type	<i>boolean</i>
Remarks	Includes the device a coin dispenser.

7.3.2.12 cylinders (R)

Type	<i>boolean</i>
Remarks	Can the coin dispenser accept a number of coins per cylinder as input or are only totals allowed.

7.3.2.13 cashBox (R)

Type	<i>boolean</i>
Remarks	Can the service handle a cash box.

7.3.2.14 refill (R)

Type	<i>boolean</i>
Remarks	Can the device be refilled by placing bills on the stack.

7.3.2.15 dispense (R)

Type	<i>boolean</i>
Remarks	Can the device dispense cash.

7.3.2.16 deposit (R)

Type	<i>boolean</i>
Remarks	Can the device deposit cash.

7.3.2.17 checkVandalism (R)

Type	<i>boolean</i>
Remarks	Can the device detect vandalism.

7.3.2.18 intermediateStacker (R)

Type	<i>boolean</i>
Remarks	Has the device a temporary storage before presenting bills.

7.3.2.19 billsTakenSensor (R)

Type	<i>boolean</i>
Remarks	Has the device a bills taken sensor.

7.3.2.20 inputPositions (R)

Type	<i>int</i>
Remarks	Specifies the possible input position to accept cash. (Defined as <i>dispense position codes</i>)

7.3.2.21 outputPositions (R)

Type	<i>int</i>
Remarks	Specifies the possible output position to dispense cash. (Defined as <i>dispense position codes</i>)

7.3.2.22 defaultInputPosition (R)

Type	<i>int</i>
Remarks	Specifies the default input position to accept cash. (Defined as <i>dispense position codes</i>)

7.3.2.23 defaultOutputPosition (R)

Type	<i>int</i>
Remarks	Specifies the default output position to dispense cash. (Defined as <i>dispense position codes</i>)

7.3.2.24 silentAlarm (R)

Type	<i>boolean</i>
Remarks	Does the device support a silent alarm feature.

7.3.2.25 escrow (R)

Type	<i>boolean</i>
Remarks	Does the device support an escrow .

7.3.2.26 escrowSize (R)

Type	<i>int</i>
Remarks	Specifies the maximum number of bills on the escrow.

7.3.2.27 detector (R)

Type
Remarks

boolean
Does the device support a detector to verify accepted cash.

7.3.2.28 baitTrap (R)

Type
Remarks

boolean
Does the device support a functionality to emit marked notes during dispense.

7.3.2.29 vendorData (R)

Type
Remarks

java.lang.String
Vendor specific data.

7.4 JxfsCashInOrder

7.4.1.1 Usage

This class specifies all data required for cash-in operations.

7.4.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
denomination	<i>JxfsDenomination</i>	RW
currency	<i>JxfsCurrency</i>	RW

Constructor	Parameter	Parameter-Type
JxfsCashInOrder	<i>denomination</i>	JxfsDenomination
	<i>currency</i>	JxfsCurrency

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.4.2 Properties

7.4.2.1 denomination (RW)

Type	<i>JxfsDenomination</i>
Remarks	Specifies the amount to cash-in or the amount accepted.

7.4.2.2 currency (RW)

Type	<i>JxfsCurrency</i>
Remarks	Specifies the currency to use.

7.5 JxfsCashType

7.5.1.1 Usage

This class is used to carry all the information that is required to uniquely define a cash item (e.g.: a bank note or coin).

7.5.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
kind	<i>int</i>	R
currencyCode	<i>JxfsCurrencyCode</i>	R
value	<i>int</i>	R
variant	<i>int</i>	R

Constructor	Parameter	Parameter-Type
JxfsCashType	<i>kind</i>	int
	<i>currencyCode</i>	JxfsCurrencyCode
	<i>value</i>	int
	<i>variant</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.5.2 Properties

7.5.2.1 kind (R)

Type	<i>int</i>
Remarks	The type of the value, a note or a coin. One of the following values: JXFS_C_CDR_CURR_BILL JXFS_C_CDR_CURR_COIN

7.5.2.2 currencyCode (R)

Type	<i>JxfsCurrencyCode</i>
Remarks	Defines the currency code for this type of cash.

7.5.2.3 value (R)

Type	<i>int</i>
Remarks	Value of bill expressed in MDUs.

7.5.2.4 variant (R)

Type	<i>int</i>
Remarks	The variant of the cash item represented. The constant JXFS_C_CDR_NO_VARIANT may be used to express that the variant information is not supported. Other values may be vendor specific.

7.6 JxfsCashUnit

7.6.1.1 Usage

Information about the status and contents of the logical and physical cash units. Each logical bill or coin type cash unit can be composed of one or more physical cash units. All counters are pure software counters. Due to this fact these values can differ from the actual physical cash counts.

7.6.1.2 Summary

Extends	Implements	
JxfsType		

Property	Type	Access
rejectCount	<i>int</i>	RW

Constructor	Parameter	Parameter-Type
JxfsCashUnit	<i>rejectCount</i>	int

Method	Return	May use after
addLogicalUnit	<i>boolean</i>	
getLogicalUnits	<i>java.util.Vector</i>	

7.6.2 Properties

7.6.2.1 rejectCount (RW)

Type	<i>int</i>
Remarks	Counter for all reject actions in the device.

7.6.3 Methods

7.6.3.1 addLogicalUnit

Syntax	<i>boolean addLogicalUnit(JxfsLogicalCashUnit logicalCashUnit)</i>		
Remarks	Add a logical cash unit.		
Parameter	Type	Name	Meaning
	<i>JxfsLogicalCashUnit</i>	logicalCashUnit	Add a logical cash unit to the internal list of cash units.

7.6.3.2 getLogicalUnits

Syntax	<i>java.util.Vector getLogicalUnits()</i>
Remarks	Returns vector of logical cash units.

7.7 JxfsCurrency

7.7.1.1 Usage

Objects of this class are used to define a supported currency. Each currency has a currency identifier (a three character code) and a currency exponent.

7.7.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
currencyCode	<i>JxfsCurrencyCode</i>	R
exponent	<i>int</i>	R

Constructor	Parameter	Parameter-Type
JxfsCurrency	<i>currencyCode</i>	JxfsCurrencyCode
	<i>exponent</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.7.2 Properties

7.7.2.1 currencyCode (R)

Type	<i>JxfsCurrencyCode</i>
Remarks	A 3-character length upper case string detailing a currency code as defined by the ISO standard, ISO 4217.

7.7.2.2 exponent (R)

Type	<i>int</i>
Remarks	JxfsCurrency exponent.

7.8 JxfCurrencyCode

7.8.1.1 Usage

Used to specify the country specific code (3-character string) for a given currency.

7.8.1.2 Summary

Extends	Implements
JxfType	

Property	Type	Access
currencyCode	<i>String</i>	R

Constructor	Parameter	Parameter-Type
JxfCurrencyCode	<i>currencyCode</i>	String

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	

7.8.2 Properties

7.8.2.1 currencyCode (R)

Type	<i>String</i>
Remarks	A 3-character length upper case string detailing a currency code as defined by the ISO standard.

7.9 JxfsDelay

7.9.1.1 Usage

A JxfsDelay object stores the time the opening of the safedoor is delayed.

7.9.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
delay	<i>int</i>	R

Constructor	Parameter	Parameter-Type
JxfsDelay	<i>delay</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.9.2 Properties

7.9.2.1 delay (R)

Type	<i>int</i>
Remarks	Specifies the time to delay in milliseconds.

7.10 JxfsDenomination

7.10.1.1 Usage

The JxfsDenomination holds a collection of JxfsDenominationItems that sum up to an amount of cash.

7.10.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
items	<i>java.lang.Vector</i>	RW
amount	<i>long</i>	RW
cashBox	<i>long</i>	RW

Constructor	Parameter	Parameter-Type
JxfsDenomination	<i>items</i>	<i>java.lang.Vector</i>
	<i>amount</i>	<i>long</i>
	<i>cashBox</i>	<i>long</i>

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	
<i>addItem</i>	<i>boolean</i>	

7.10.2 Properties

7.10.2.1 items (RW)

Type	<i>java.lang.Vector</i>
Remarks	A list of <i>JxfsDenominationItems</i> .
Note for <i>denominate</i>	These items define the asset used for denominate.

7.10.2.2 amount (RW)

Type	<i>long</i>
Remarks	Amount expressed in MDUs.
Note for <i>denominate</i>	This is the amount to be denominated.

7.10.2.3 cashBox (RW)

Type	<i>long</i>
Remarks	Cashbox amount expressed in MDUs.
Note for <i>denominate</i>	On return of the denominate-operation, this defines an amount, that could not be denominated.

7.10.3 Methods

7.10.3.1 addItem

Syntax	<i>boolean addItem(JxfsDenominationItem item)</i>		
Remarks	Add a <i>JxfsDenominationItem</i> to this denomination.		
Parameter	Type	Name	Meaning
	<i>JxfsDenominationItem</i>	item	

7.11 JxfsDenominationItem

7.11.1.1 Usage

A JxfsDenominationItem specifies a logical cash unit and the number of bills or coins that were dispensed from this unit or that should be deposited into this unit.

7.11.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
unit	<i>int</i>	RW
count	<i>int</i>	RW

Constructor	Parameter	Parameter-Type
JxfsDenominationItem	<i>unit</i>	int
	<i>count</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.11.2 Properties

7.11.2.1 unit (R)

Type	<i>int</i>
Remarks	Number of logical cash unit.

7.11.2.2 count (R)

Type	<i>int</i>
Remarks	Number of bills/coins to dispense/deposit.

7.12 JxfsDispenseOrder

7.12.1.1 Usage

This class specifies all data required for dispense operations.

7.12.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
orderID	<i>int</i>	RW
queueID	<i>int</i>	RW
denomination	<i>JxfsDenomination</i>	RW
currency	<i>JxfsCurrency</i>	RW
when	<i>java.util.Date</i>	RW
delay	<i>long</i>	RW

Constructor	Parameter	Parameter-Type
JxfsDispenseOrder	<i>orderID</i>	int
	<i>queueID</i>	int
	<i>denomination</i>	JxfsDenomination
	<i>currency</i>	JxfsCurrency
	<i>when</i>	java.util.Date
	<i>delay</i>	long

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.12.2 Properties

7.12.2.1 orderID (RW)

Type *int*
Remarks Used to identify a dispense order.

7.12.2.2 queueID (RW)

Type *int*
Remarks Specifies the queue the dispense order was inserted.
One of the following values: (UVV Delayed Order Queue codes)
JXFS_C_CDR_DO_DELAYED
JXFS_C_CDR_DO_LAQ

7.12.2.3 denomination (RW)

Type *JxfsDenomination*
Remarks Specifies the amount of cash to dispense.

7.12.2.4 currency (RW)

Type *JxfsCurrency*
Remarks Specifies the currency to use.

7.12.2.5 when (RW)

Type	<i>java.util.Date</i>
Remarks	Time the dispense was requested.

7.12.2.6 delay (RW)

Type	<i>long</i>
Remarks	Delay in ms from <i>when</i> . If <i>delay</i> equals 0, then the dispense order was processed immediately, else, if <i>delay</i> is greater 0, then the order is delayed for <i>delay</i> milliseconds.

7.13 JxfsDispenseRequest

7.13.1.1 Usage

This class specifies all data required for a dispense request.

7.13.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
mixNumber	<i>int</i>	RW
denomination	<i>JxfsDenomination</i>	RW
currency	<i>JxfsCurrency</i>	RW
position	<i>int</i>	RW

Constructor	Parameter	Parameter-Type
JxfsDispenseRequest	<i>mixNumber</i>	int
	<i>denomination</i>	JxfsDenomination
	<i>currency</i>	JxfsCurrency
	<i>position</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.13.2 Properties

7.13.2.1 mixNumber (RW)

Type	<i>int</i>
Remarks	Type of denomination algorithm to use.

7.13.2.2 denomination (RW)

Type	<i>JxfsDenomination</i>
Remarks	Specifies the amount of cash to dispense.

7.13.2.3 currency (RW)

Type	<i>JxfsCurrency</i>
Remarks	Specifies the currency to use.

7.13.2.4 position (RW)

Type	<i>int</i>
Remarks	Specifies the output position to use for presenting money.

7.14 JxfsLogicalCashUnit

7.14.1.1 Usage

Logical information about a cash unit. Each logical unit can be composed of multiple physical units.

7.14.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
cashType	<i>JxfsCashType</i>	RW
number	<i>int</i>	RW
cuKind	<i>int</i>	RW
cuType	<i>int</i>	RW
unitID	<i>java.lang.String</i>	RW
initialCount	<i>int</i>	RW
count	<i>int</i>	RW
threshold	<i>JxfsThreshold</i>	RW
appLock	<i>boolean</i>	RW
devLock	<i>boolean</i>	RW
status	<i>int</i>	RW
thresholdStatus	<i>JxfsThresholdStatus</i>	RW
physicalName	<i>java.lang.String</i>	RW
physicalUnits	<i>java.util.Vector</i>	RW
depositCount	<i>int</i>	RW
dispenseCount	<i>int</i>	RW
rejectCount	<i>int</i>	RW

Constructor	Parameter	Parameter-Type
JxfsLogicalCashUnit	<i>cashType</i>	JxfsCashType
	<i>number</i>	int
	<i>cuKind</i>	int
	<i>cuType</i>	int
	<i>unitID</i>	java.lang.String
	<i>initialCount</i>	int
	<i>count</i>	int
	<i>threshold</i>	JxfsThreshold
	<i>appLock</i>	boolean
	<i>devLock</i>	boolean
	<i>status</i>	int
	<i>thresholdStatus</i>	JxfsThresholdStatus
	<i>physicalName</i>	java.lang.String
	<i>physicalUnits</i>	java.util.Vector
	<i>depositCount</i>	int
	<i>dispenseCount</i>	int
	<i>rejectCount</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	
<i>addUnit</i>	<i>boolean</i>	

7.14.2 Properties

7.14.2.1 cashType (RW)

Type	<i>JxfsCashType</i>
Remarks	Defines the type of cash used by this cash unit.

7.14.2.2 number (RW)

Type	<i>int</i>
Remarks	Logical number of cash unit. Starting with a value of one (1) for the first cash unit. Incremented by one for the next units.

7.14.2.3 cuKind (RW)

Type	<i>int</i>
Remarks	Specifies, if cash unit can dispense, deposit cash or both. One of the following values: JXFS_C_CDR_LCU_NA JXFS_C_CDR_LCU_DISPENSE JXFS_C_CDR_LCU_DEPOSIT JXFS_C_CDR_LCU_RECYCLE

7.14.2.4 cuType (RW)

Type	<i>int</i>
Remarks	Type of cash unit. One of the following values: JXFS_C_CDR_LCU_NA JXFS_C_CDR_LCU_REJECT_CASSETTE JXFS_C_CDR_LCU_OVERFLOW_CASSETTE JXFS_C_CDR_LCU_BILL_CASSETTE JXFS_C_CDR_LCU_COIN_CYLINDER JXFS_C_CDR_LCU_COIN_DISPENSER JXFS_C_CDR_LCU_RETRACT_CASSETTE JXFS_C_CDR_LCU_COUPON JXFS_C_CDR_LCU_DOCUMENT JXFS_C_CDR_LCU_ESCROW JXFS_C_CDR_LCU_BAIT_TRAP

7.14.2.5 unitID (RW)

Type	<i>java.lang.String</i>
Remarks	Identification value for a cash unit.

7.14.2.6 initialCount (RW)

Type	<i>int</i>
Remarks	Initial number of coins or bills. This value is persistent and can be modified by power failures, opens, closes and system resets. It is set by <i>endExchange</i> and <i>updateCashUnit</i> .

7.14.2.7 count (RW)

Type	<i>int</i>
Remarks	Actual count of coins or bills. This value is persistent and can be modified by power failures, opens, closes and system resets. It is set by <i>endExchange()</i> and <i>updateCashUnit()</i> . It will be adjusted by dispense or deposit actions.
Note	If this is a reject cassette, this value gives the number of rejects. If this is a retract cassette, this value gives the numbers of retracts.

7.14.2.8 threshold (RW)

Type	<i>JxfsThreshold</i>
Remarks	Specifies the threshold values for a cash unit.

7.14.2.9 appLock (RW)

Type	<i>boolean</i>
Remarks	If set to TRUE, the cash unit is locked by the application and can not be used until unlocked by the application.

7.14.2.10 devLock (RW)

Type	<i>boolean</i>
Remarks	If set to TRUE, the cash unit is locked by the device and can not be used until unlocked by the device service.

7.14.2.11 status (RW)

Type	<i>int</i>
Remarks	Cash unit status. One of the following values: JXFS_C_CDR_LCU_UNKNOWN JXFS_C_CDR_LCU_OK JXFS_C_CDR_LCU_INOP JXFS_C_CDR_LCU_MISSING JXFS_C_CDR_LCU_NO_VALUE JXFS_C_CDR_LCU_NO_REF JXFS_C_CDR_LCU_NOT_DISPENSEABLE

7.14.2.12 thresholdStatus (RW)

Type	<i>JxfsThresholdStatus</i>
Remarks	Cash unit threshold status.

7.14.2.13 physicalName (RW)

Type	<i>java.lang.String</i>
Remarks	Name of the physical location of the cash unit in the dispenser device. This field is only used when logical unit equals physical unit.

7.14.2.14 physicalUnits (RW)

Type	<i>java.util.Vector</i>
Remarks	Return vector of physical cash units.

7.14.2.15 depositCount (RW)

Type	<i>int</i>
Remarks	Number of bills, that were deposited.

7.14.2.16 dispenseCount (RW)

Type	<i>int</i>
Remarks	Number of bills, that were dispensed.

7.14.2.17 rejectCount (RW)

Type	<i>int</i>
Remarks	Counter for reject actions in the device.

7.14.3 Methods

7.14.3.1 addUnit

Syntax	<i>boolean addUnit(JxfsPhysicalCashUnit unit)</i>		
Remarks	Add a <i>JxfsPhysicalCashUnit</i> to this logical cash unit.		
Parameter	Type	Name	Meaning
	<i>JxfsPhysicalCashUnit</i>	unit	

7.15 JxfsMixEntry

7.15.1.1 Usage

One entry in a JxfsMixItem. It contains a reference to the logical cash unit and the number of bills/coins used in mixing.

7.15.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
lcu	<i>int</i>	R
count	<i>int</i>	R

Constructor	Parameter	Parameter-Type
JxfsMixEntry	<i>lcu</i>	int
	<i>count</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.15.2 Properties

7.15.2.1 lcu (R)

Type	<i>int</i>
Remarks	Number of logical cash unit.

7.15.2.2 count (R)

Type	<i>int</i>
Remarks	Number of bills or coins.

7.16 JxfsMixInfo

7.16.1.1 Usage

Type for identifying mix algorithm and/or house mix tables.

7.16.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
number	<i>int</i>	R
mixType	<i>int</i>	R
mixAlgorithmType	<i>int</i>	R
name	<i>java.lang.String</i>	R

Constructor	Parameter	Parameter-Type
JxfsMixInfo	<i>number</i>	int
	<i>mixType</i>	int
	<i>mixAlgorithmType</i>	int
	<i>name</i>	java.lang.String

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.16.2 Properties

7.16.2.1 number (R)

Type	<i>int</i>
Remarks	Number of this mixtype item.

7.16.2.2 mixType (R)

Type	<i>int</i>
Remarks	Specifies that an algorithm, a mix table or the current denomination should be used. If no mix algorithm is defined, then the following field is not used. These values may be or'ed. One or a combination of the following values: JXFS_C_CDR_MIX_ALGORITHM JXFS_C_CDR_MIX_TABLE JXFS_C_CDR_MIX_DENOM

7.16.2.3 mixAlgorithmType (R)

Type	<i>int</i>
Remarks	This selects the type of algorithm or mix table. One of the following values: JXFS_C_CDR_MXA_MIN_BILLS JXFS_C_CDR_MXA_EQUAL_EMPTY

7.16.2.4 name (R)

Type	<i>java.lang.String</i>
Remarks	Name of algorithm or mix table.

7.17 JxfsMixItem

7.17.1.1 Usage

Specifies an amount used in a JxfsMixTable. (In minimum dispense units, mdu). It also contains a list of entries that specify the logical cash units and the number of bills/coins used.

7.17.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
amount	<i>long</i>	R
entries	<i>Vector</i>	R

Constructor	Parameter	Parameter-Type
JxfsMixItem	<i>amount</i>	long
	<i>entries</i>	Vector

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.17.2 Properties

7.17.2.1 amount (R)

Type	<i>long</i>
Remarks	Amount used in the mix table in mdu's.

7.17.2.2 entries (R)

Type	<i>Vector of JxfsMixEntry's</i>
Remarks	List of JxfsMixEntry's.

7.18 JxfsMixTable

7.18.1.1 Usage

Contains complete description of a mix table.

7.18.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
mixInfo	<i>JxfsMixInfo</i>	RW
items	<i>Vector</i>	RW

Constructor	Parameter	Parameter-Type
JxfsMixTable	<i>mixInfo</i>	JxfsMixInfo
	<i>items</i>	Vector

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.18.2 Properties

7.18.2.1 mixInfo (RW)

Type	<i>JxfsMixInfo</i>
Remarks	Identification of mix algorithm and/or house mix table.

7.18.2.2 items (RW)

Type	<i>Vector of JxfsMixItem's</i>
Remarks	Specifies amount used in the JxfsMixTable.

7.19 JxfsPhysicalCashUnit

7.19.1.1 Usage

Information about a physical cash unit.

7.19.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
name	<i>java.lang.String</i>	R
unitID	<i>java.lang.String</i>	R
count	<i>int</i>	R
threshold	<i>JxfsThreshold</i>	R
status	<i>int</i>	R
thresholdStatus	<i>JxfsThresholdStatus</i>	R
lock	<i>boolean</i>	R

Constructor	Parameter	Parameter-Type
JxfsPhysicalCashUnit	<i>name</i>	<i>java.lang.String</i>
	<i>unitID</i>	<i>java.lang.String</i>
	<i>count</i>	<i>int</i>
	<i>threshold</i>	<i>JxfsThreshold</i>
	<i>status</i>	<i>int</i>
	<i>thresholdStatus</i>	<i>JxfsThresholdStatus</i>
	<i>lock</i>	<i>boolean</i>

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.19.2 Properties

7.19.2.1 name (R)

Type	<i>java.lang.String</i>
Remarks	Name of the physical location in the dispenser device where this cash unit is installed.

7.19.2.2 unitID (R)

Type	<i>java.lang.String</i>
Remarks	Cash unit ID.

7.19.2.3 count (R)

Type	<i>int</i>
Remarks	Actual count of bills or coins in the physical cash unit.

7.19.2.4 threshold (R)

Type	<i>JxfsThreshold</i>
Remarks	Specifies the threshold values for a cash unit.

7.19.2.5 status (R)

Type	<i>int</i>
Remarks	Status of the physical cash unit. May have the same range of values as LogicalCashUnit.status.

7.19.2.6 thresholdStatus (R)

Type	<i>JxfsThresholdStatus</i>
Remarks	Thresholdstatus of the physical cash unit.

7.19.2.7 lock (R)

Type	<i>boolean</i>
Remarks	Lock status of the physical cash unit. Can be used from application and device service. Usually used for hot swap of cassettes.

7.20 JxfsRetractArea

7.20.1.1 Usage

To be defined

7.20.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access

Constructor	Parameter	Parameter-Type
JxfsRetractArea	<i>none</i>	

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.20.2 Properties

7.20.2.1 xxx (R)

Type	XXX
Remarks	none

7.21 JxfsThreshold

7.21.1.1 Usage

Defines limits for cassettes.

7.21.1.2 Summary

Extends	Implements
JxfsType	

Property	Type	Access
full	<i>int</i>	R
high	<i>int</i>	R
low	<i>int</i>	R
empty	<i>int</i>	R

Constructor	Parameter	Parameter-Type
JxfsThreshold	<i>full</i>	int
	<i>high</i>	int
	<i>low</i>	int
	<i>empty</i>	int

Method	Return	May use after
<i>getProperty</i>	<i>Property</i>	
<i>setProperty</i>	<i>Property</i>	

7.21.2 Properties

7.21.2.1 full (R)

Type *int*
Remarks Specifies the full level for the cash unit.

7.21.2.2 high (R)

Type *int*
Remarks Specifies the high level for the cash unit.

7.21.2.3 low (R)

Type *int*
Remarks Specifies the low level for the cash unit.

7.21.2.4 empty (R)

Type *int*
Remarks Specifies the empty level for the cash unit.

8 Error Codes

Value	Meaning
JXFS_E_CDR_CASHIN_ACTIVE	The device has already a cashInStart command issued.
JXFS_E_CDR_CASH_DEVICE_ERROR	An unspecified error occurred.
JXFS_E_CDR_CASH_UNIT_ERROR	A selected cash unit caused an error.
JXFS_E_CDR_CMD_NOT_SUPPORTED	Command is not supported.
JXFS_E_CDR_EXCHANGE_ACTIVE	The device is in an exchange state.
JXFS_E_CDR_DELAYED_DISPENSE	Dispense order is delayed.
JXFS_E_CDR_ILLEGAL_DISPENSE_ORDER	Invalid orderID during dispenseExec.
JXFS_E_CDR_INPUT_REFUSED	CashIn operation failure.
JXFS_E_CDR_INVALID_CASH_UNIT	Invalid cash unit ID.
JXFS_E_CDR_INVALID_CURRENCY	JxfsCurrency type is not configured.
JXFS_E_CDR_INVALID_BILL	Unvalid bill detected during cashin.
JXFS_E_CDR_INVALID_COIN	Unvalid coin detected during cashin.
JXFS_E_CDR_INVALID_DENOMINATION	The sum values for cashbox, cash unit and coins were greater than the amount specified.
JXFS_E_CDR_INVALID_MIXNUMBER	The number refers to a defined and reserved mix algorithm.
JXFS_E_CDR_INVALID_RETRACT	Retract method is invalid for this system.
JXFS_E_CDR_NOT_DISPENSABLE	The amount is not dispensable.
JXFS_E_CDR_NO_BILLS	There were no bills on the stacker to present.
JXFS_E_CDR_NO_CASHIN_STARTED	cashInStart was not called.
JXFS_E_CDR_NO_EXCHANGE_ACTIVE	The device is not in an exchange state.
JXFS_E_CDR_RESET_REQUIRED	Reset operation is required.
JXFS_E_CDR_TOO_MANY_BILLS	The request would require too many bills to be dispensed.
JXFS_E_CDR_TOO_MANY_COINS	The request would require too many coins to be dispensed.
JXFS_E_CDR_UNABLE_OPEN_SHUTTER	Shutter could not be opened.
JXFS_E_CDR_UVV_IN_PROCESS	UVV delay is still active for this order.

9 Exception Codes

Value	Meaning
JXFS_E_CLAIMED	Device is already claimed.
JXFS_E_CLOSED	Device has not been opened yet.
JXFS_E_INVALID_PARAMETER	An invalid parameter was given to the operation.
JXFS_E_NOT_CLAIMED	Device is not claimed by caller.
JXFS_E_NOT_SUPPORTED	Operation is not supported by device.
JXFS_E_NO_HARDWARE	No Device is connected to the workstation.
JXFS_E_OFFLINE	Device is offline.
JXFS_E_REMOTE	Communication error during remote call.
JXFS_E_TIMEOUT	A timeout has occurred.

10 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.

10.1 JxfsCdrStatus

Extends	Implements
JxfsStatus	

Query	Return
getCashTrayStatus	JxfsCashTrayStatus
getCashUnitStatus	JxfsCashUnitStatus
getDeviceStatus	JxfsDeviceStatus
getDispenserStatus	JxfsDispenserStatus
getDispenseOrderStatus	JxfsDispenseOrderStatus
getIntermediateStackerStatus	JxfsIntermediateStackerStatus
getPresentStatus	JxfsPresentStatus
getSafeDoorStatus	JxfsSafeDoorStatus
getShutterStatus	JxfsShutterStatus
getTransportStatus	JxfsTransportStatus
getVandalismStatus	JxfsVandalismStatus

10.2 JxfsCashTrayStatus

Extends	Implements
JxfsType	

Query	Return
isEmpty	<i>boolean</i>
isNotEmpty	<i>boolean</i>
isNotSupported	<i>boolean</i>
isUnknown	<i>boolean</i>

10.3 JxfsCashUnitStatus

Extends	Implements
JxfsType	

Query	Return
getCashUnit	<i>JxfsCashUnit</i>

10.4 JxfsDeviceStatus

Extends	Implements
JxfsType	

Query	Return
isOnLine	<i>boolean</i>
isOffLine	<i>boolean</i>
isPowerOff	<i>boolean</i>
isBusy	<i>boolean</i>
isNoDevice	<i>boolean</i>
isUserError	<i>boolean</i>
isHardwareError	<i>boolean</i>

10.5 JxfsDispenserStatus

Extends	Implements
JxfsType	

Query	Return
isOk	<i>boolean</i>
isJxfsCashUnitState	<i>boolean</i>
isJxfsCashUnitStop	<i>boolean</i>
isJxfsCashUnitUnknown	<i>boolean</i>

10.6 JxfsDispenseOrderStatus

Extends	Implements
JxfsType	

Query	Return
getDispenseOrder	<i>JxfsDispenseOrder</i>

10.7 JxfsIntermediateStackerStatus

Extends	Implements
JxfsType	

Query	Return
isEmpty	<i>boolean</i>
isNotEmpty	<i>boolean</i>
isUnknown	<i>boolean</i>
isNotSupported	<i>boolean</i>

10.8 JxfsPresentStatus

Extends	Implements
JxfsType	

Query	Return
isUnknown	<i>boolean</i>
isPresented	<i>boolean</i>

10.9 JxfsSafeDoorStatus

Extends	Implements

JxfsType	
Query	Return
isNotSupported	<i>boolean</i>
isOpen	<i>boolean</i>
isClosed	<i>boolean</i>
isLocked	<i>boolean</i>
isUnknown	<i>boolean</i>

10.10 JxfsShutterStatus

Extends	Implements
JxfsType	
Query	Return
isClosed	<i>boolean</i>
isOpen	<i>boolean</i>
isJammed	<i>boolean</i>
isNotSupported	<i>boolean</i>
isUnknown	<i>boolean</i>

10.11 JxfsThresholdStatus

Extends	Implements
JxfsType	
Query	Return
isFull	<i>boolean</i>
isHigh	<i>boolean</i>
isLow	<i>boolean</i>
isEmpty	<i>boolean</i>
isUnknown	<i>boolean</i>

10.12 JxfsTransportStatus

Extends	Implements
JxfsType	
Query	Return
isOk	<i>boolean</i>
isInOp	<i>boolean</i>
isNotSupported	<i>boolean</i>
isUnknown	<i>boolean</i>

10.13 JxfsVandalismStatus

Extends	Implements
JxfsType	
Query	Return
isManipulation	<i>boolean</i>

11 Status Codes

11.1.1.1 General Status codes

General status codes that specify a value change.

Value	Meaning
JXFS_S_CDR_CASHUNIT_CHANGED	
JXFS_S_CDR_CASHUNIT_CONFIGURATION_CHANGED	
JXFS_S_CDR_CASHUNIT_THRESHOLD	
JXFS_S_CDR_DELAYED_DISPENSE	
JXFS_S_CDR_RETRACT_AREA_CHANGED	
JXFS_S_CDR_SHUTTER_CHANGED	
JXFS_S_CDR_ORDER_CHANGED	
JXFS_S_CDR_CASH_TRAY_CHANGED	
JXFS_S_CDR_DISPENSER_STATUS_CHANGED	
JXFS_S_CDR_DEVICE_STATUS_CHANGED	
JXFS_S_CDR_INTERMEDIATE_STACKER_CHANGED	
JXFS_S_CDR_PRESENT_CHANGED	
JXFS_S_CDR_SAFEDOOR_CHANGED	
JXFS_S_CDR_SHUTTER_CHANGED	
JXFS_S_CDR_TRANSPORT_CHANGED	
JXFS_S_CDR_VANDALISM_CHANGED	
JXFS_S_CDR_DATE_TIME_CHANGED	

11.1.1.2 Cash Tray Status codes

Defines the possible states a cash tray can report.

Value	Meaning
JXFS_S_CDR_CT_EMPTY	The cash tray is empty
JXFS_S_CDR_CT_NOT_EMPTY	The cash tray is not empty
JXFS_S_CDR_CT_NOT_SUPPORTED	The physical device has no cash tray or cash tray state reporting is not supported
JXFS_S_CDR_CT_UNKNOWN	Due to a hardware error or other condition the state of the cash tray cannot be determined

11.1.1.3 Device Status codes

Reports the state of the cash dispenser/recycler device.

Value	Meaning
JXFS_S_CDR_DS_ON_LINE	The device is online
JXFS_S_CDR_DS_OFF_LINE	The device is offline
JXFS_S_CDR_DS_POWER_OFF	The device is present, but is powered off
JXFS_S_CDR_DS_BUSY	The device is processing a request
JXFS_S_CDR_DS_NO_DEVICE	There is no device connected
JXFS_S_CDR_DS_USER_ERROR	The device is present, but a person is preventing proper device operation
JXFS_S_CDR_DS_HARDWARE_ERROR	The device is present and powered on, but is inoperable due to a hardware error

11.1.1.4 Dispenser Status codes

Specifies the state of the dispenser/recycler cash units device.

Value	Meaning
JXFS_S_CDR_DIS_OK	All logical cash units are ok.
JXFS_S_CDR_DIS_CU_STATE	One of the logical cash units present is in an abnormal state. The dispenser is operational, but one or more of the cash units is in a low, empty or inoperative condition. Bills can still be dispensed from at least one of the cash units.
JXFS_S_CDR_DIS_CU_STOP	Due to a cash unit failure dispensing is impossible. The dispenser is operational, but no bills can be dispensed because all of the cash units are in an empty or inoperative condition. This state occurs when a reject cash unit is full or no reject cassette is present.
JXFS_S_CDR_DIS_CU_UNKNOWN	Due to a hardware error or other condition, the state of the cash units cannot be determined.

11.1.1.5 Intermediate Stacker Status codes

Specifies the state of the intermediate stacker.

Value	Meaning
JXFS_S_CDR_IS_EMPTY	The intermediate stacker is empty.
JXFS_S_CDR_IS_NOT_EMPTY	The intermediate stacker is not empty.
JXFS_S_CDR_IS_UNKNOWN	Due to a hardware error or other condition, the state of the intermediate stacker cannot be determined.
JXFS_S_CDR_IS_NOT_SUPPORTED	The physical device has no intermediate stacker.

11.1.1.6 Present Status codes

Reports the state of the last dispense. ATM only.

Value	Meaning
JXFS_S_CDR_PR_UNKNOWN	It is unknown if the money could be accessed by the customer.
JXFS_S_CDR_PR_NOT_PRESENTED	The money was not presented.
JXFS_S_CDR_PR_PRESENTED	The money was presented. This value is set as soon as the bills are accessible by the customer.
JXFS_S_CDR_CASH_TAKEN	The cash was taken by the user.

11.1.1.7 Safe Door Status codes

Reports the state of the safe door.

Value	Meaning
JXFS_S_CDR_SD_NOT_SUPPORTED	The physical device has no safe door or door state reporting is not supported.
JXFS_S_CDR_SD_OPEN	The safe door is open.
JXFS_S_CDR_SD_CLOSED	The safe door is closed but not locked.
JXFS_S_CDR_SD_LOCKED	The safe door is closed and locked.
JXFS_S_CDR_SD_UNKNOWN	Due to a hardware error or other condition, the state of the door cannot be determined.

11.1.1.8 Shutter Status codes

State of special security rules which must be observed during dispense of cash.

Value	Meaning
JXFS_S_CDR_SHT_CLOSED	The shutter is closed.
JXFS_S_CDR_SHT_OPEN	The shutter is open.
JXFS_S_CDR_SHT_JAMMED	The shutter is jammed.
JXFS_S_CDR_SHT_NOT_SUPPORTED	The physical device has no shutter or shutter state reporting is not supported.
JXFS_S_CDR_SHT_UNKNOWN	Due to a hardware error or other condition, the state of the shutter cannot be determined.

11.1.1.9 Transport Status codes

Reports the state of the transport mechanism.

Value	Meaning
JXFS_S_CDR_TP_OK	The transport is in a good state.
JXFS_S_CDR_TP_INOP	Due to a hardware failure or media jam, the transport is inoperable.
JXFS_S_CDR_TP_NOT_SUPPORTED	The physical device has no transport or transport state reporting is not supported.
JXFS_S_CDR_TP_UNKNOWN	Due to a hardware error or other condition, the state of the transport cannot be determined.

11.1.1.10 Vandalism Status codes

Reports the state of the vandalism checker.

Value	Meaning
JXFS_S_CDR_VAN_MANIPULATION	An attempt to manipulated the device has been detected.
JXFS_S_CDR_VAN_NO_MANIPULATION	No attempt has been made to manipulate the device.

12 Intermediate Event Codes

Value	Meaning
JXFS_I_CDR_CASH_AVAILABLE	Cash is available at the delivery slot.
JXFS_I_CDR_CASH_TAKEN	Cash is removed from the delivery slot.
JXFS_I_CDR_DELAYED_ORDER_READY	An order is ready for dispense.
JXFS_I_CDR_PARTIAL_DISPENSE	A partial dispense order is ready.
JXFS_I_CDR_SAFE_DOOR_LOCKED	Safedoor is locked.
JXFS_I_CDR_SAFE_DOOR_UNLOCKED	Safedoor is unlocked.

13 Constants

13.1.1.1 Dispense position codes

Following dispense position codes can be or'ed groupwise. This is possible for a capability query.

Value	Meaning
JXFS_C_CDR_POS_NONE	No position selected
JXFS_C_CDR_POS_DEFAULT	Use configured position
JXFS_C_CDR_POS_LEFT	Dispense to left side
JXFS_C_CDR_POS_CENTER	Dispense to the middle
JXFS_C_CDR_POS_RIGHT	Dispense to right side

Value	Meaning
JXFS_C_CDR_POS_OVERFLOW	Dispense to overflow cassette
JXFS_C_CDR_POS_REJECT	Dispense to reject cassette

Value	Meaning
JXFS_C_CDR_POS_TOP	Used for shutter.
JXFS_C_CDR_POS_BOTTOM	Used for shutter.

13.1.1.2 Device Type codes

Following device type codes can be or'ed.

Value	Meaning
JXFS_C_CDR_TYPE_NONE	Device is not defined
JXFS_C_CDR_TYPE_DISPENSER	Device is a Cash Dispenser
JXFS_C_CDR_TYPE_RECYCLER	Device is a Cash Recycler
JXFS_C_CDR_TYPE_ATM	Device is a Automated Teller Machine

13.1.1.3 Cash Type codes

Value	Meaning
JXFS_C_CDR_CURR_BILL	Item represents a bill
JXFS_C_CDR_CURR_COIN	Item represents a coin

13.1.1.4 CashUnit Kind codes

Value	Meaning
JXFS_C_CDR_LCU_DISPENSE	Cash unit can be used for dispense.
JXFS_C_CDR_LCU_DEPOSIT	Cash unit can be used for deposit.
JXFS_C_CDR_LCU_RECYCLE	Cash unit can be used for dispense and deposit.

13.1.1.5 CashUnit Type codes

Value	Meaning
JXFS_C_CDR_LCU_NA	Not applicable; cash unit is missing
JXFS_C_CDR_LCU_REJECT_CASSETTE	Reject cassette of cash dispenser
JXFS_C_CDR_LCU_OVERFLOW_CASSETTE	Overflow cassette of cash dispenser
JXFS_C_CDR_LCU_BILL_CASSETTE	Bill cassette of cash dispenser
JXFS_C_CDR_LCU_COIN_CYLINDER	Cylinder of the coin dispenser
JXFS_C_CDR_LCU_COIN_DISPENSER	Coin dispenser as a whole unit
JXFS_C_CDR_LCU_RETRACT_CASSETTE	Retract cassette of cash dispenser
JXFS_C_CDR_LCU_COUPON	Cassette for coupons or advertising materials
JXFS_C_CDR_LCU_DOCUMENT	Cassette for documents
JXFS_C_CDR_LCU_ESCROW	Cassette is an escrow
JXFS_C_CDR_LCU_BAIT_TRAP	Cash unit has bait trap capability.

13.1.1.6 CashUnit Status codes

Value	Meaning
JXFS_C_CDR_LCU_UNKNOWN	The state of the cash unit is unknown.
JXFS_C_CDR_LCU_OK	The cash unit is in a good state.
JXFS_C_CDR_LCU_FULL	The reject or retract cassette is full.
JXFS_C_CDR_LCU_HIGH	The reject or retract cassette is almost full (threshold).
JXFS_C_CDR_LCU_LOW	The reject or retract cassette is almost empty (threshold).
JXFS_C_CDR_LCU_EMPTY	The cassette or coin cylinder is empty.
JXFS_C_CDR_LCU_INOP	The cassette or coin cylinder is inoperative.
JXFS_C_CDR_LCU_MISSING	The cassette or coin cylinder is missing.
JXFS_C_CDR_LCU_NO_VALUE	The values of the specified cash unit are not available. This could happen to be, if the cassette was changed without J/XFS calls.
JXFS_C_CDR_LCU_NO_REF	There is no reference value available for the notes in this cassette.
JXFS_C_CDR_LCU_NOT_DISPENSEABLE	Cannot dispense from this cassette.

13.1.1.7 Mix Type codes

Value	Meaning
JXFS_C_CDR_MIX_ALGORITHM	A algorithm is selected for mixing
JXFS_C_CDR_MIX_TABLE	A table is selected for mixing
JXFS_C_CDR_MIX_DENOM	The current selected JxfsDenomination is used

13.1.1.8 Mix Algorithm Type codes

Value	Meaning
JXFS_C_CDR_MXA_MIN_BILLS	The minimal number of bills is used
JXFS_C_CDR_MXA_EQUAL_EMPTY	All cash units are equally emptied

13.1.1.9 UVV Delayed Order Queue codes

Value	Meaning
JXFS_C_CDR_DO_DISPENSABLE	Orders ready for processing.
JXFS_C_CDR_DO_DELAYED	All orders in delay queue.
JXFS_C_CDR_DO_LAQ	All orders in Large Amount Queue.
JXFS_C_CDR_DO_ALL	All orders in all queues.

14 Operation ID Codes

Following codes specify the operation which generated the OperationCompleteEvent.

14.1.1.1 Cash Dispenser

Value	Method
JXFS_O_CDR_DENOMINATE	<i>denominate</i>
JXFS_O_CDR_DISPENSE	<i>dispense</i>
JXFS_O_CDR_DISPENSE_EXEC	<i>dispenseExec</i>
JXFS_O_CDR_START_EXCHANGE	<i>startExchange</i>
JXFS_O_CDR_END_EXCHANGE	<i>endExchange</i>
JXFS_O_CDR_OPEN_SAFE_DOOR	<i>openSafeDoor</i>
JXFS_O_CDR_CALIBRATE_CASH_UNIT	<i>calibrateCashUnit</i>
JXFS_O_CDR_QUERY_CASH_UNIT	<i>queryCashUnit</i>
JXFS_O_CDR_QUERY_ORDER	<i>queryOrder</i>
JXFS_O_CDR_REMOVE_ORDER	<i>removeOrder</i>
JXFS_O_CDR_RESET	<i>reset</i>
JXFS_O_CDR_UPDATE_CASH_UNIT	<i>updateCashUnit</i>
JXFS_O_CDR_GET_DATE_TIME	<i>getDateTime</i>
JXFS_O_CDR_SET_DATE_TIME	<i>setDateTime</i>

14.1.1.2 Cash Recycler

Value	Method
JXFS_O_CDR_CASH_IN_START	<i>cashInStart</i>
JXFS_O_CDR_CASH_IN	<i>cashIn</i>
JXFS_O_CDR_CASH_IN_END	<i>cashInEnd</i>
JXFS_O_CDR_CASH_IN_ROLLBACK	<i>cashInRollback</i>
JXFS_O_CDR_EMPTY	<i>empty</i>

14.1.1.3 ATM

Value	Method
JXFS_O_CDR_PRESENT	<i>present</i>
JXFS_O_CDR_REJECT	<i>reject</i>
JXFS_O_CDR_RETRACT	<i>retract</i>
JXFS_O_CDR_SHUTTER_MOVE	<i>shutterMove</i>