

CEN

CWA 14923-7

WORKSHOP

May 2004

AGREEMENT

ICS 35.240.40

Supersedes CWA 13937-7:2003

English version

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part 7: Alarm Device Interface - Programmer's Reference

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

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

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

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

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



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

Management Centre: rue de Stassart, 36 B-1050 Brussels

© 2004 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.:CWA 14923-7:2004 E

Contents

Contents.....	2
Foreword.....	3
History.....	4
1 Scope.....	5
2 Overview.....	6
3 Classes and Interfaces.....	7
3.1 Class Diagram.....	8
3.2 Class and Interface Details.....	9
3.2.1 Access to properties.....	9
3.2.2 Exceptions.....	9
3.3 IJxfsAlarmControl Summary.....	10
3.4 IJxfsAlarmControl Methods.....	11
3.4.1 alarm.....	11
4 Support Classes.....	12
5 Status Event Classes.....	12
6 Codes.....	13
6.1 Operation Codes.....	13
6.1.1 IJxfsAlarmControl.....	13
6.2 Exception Codes.....	13
6.2.1 Exception Code Summary and Description.....	13
6.2.2 IJxfsAlarmControl Exception Codes.....	14
6.3 Error Codes.....	15
6.3.1 Common Codes for all operations.....	15
6.3.2 Error Code Summary and Description.....	15
7 Constants.....	16
7.1 Handling of <i>null</i> parameters.....	16
7.2 Handling of <i>null</i> return values.....	16
8 APPENDIX A : CEN/ISSS WORKSHOP 14923:2004 CORE MEMBERS :.....	16

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 2002-09-25/26 in Barcelona and a subsequent electronic review by the Workshop participants, and the final version was sent to CEN for publication on 2002-12-06.

The specification is continuously reviewed and commented in the CEN/ISSS J/XFS Workshop. 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/cenorm/businessdomains/businessdomains/informationstandardizationsystem/applying+technologies/j-xfs+workshop/index.asp>).

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 J/XFS Workshop Secretariat hosted in CEN/ISSS (jxfs-helpdesk@cenorm.be).

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
- Part 11: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Camera Specification - Programmer's Reference
- Part 12: J/eXtensions for Financial Services (J/XFS) for the Java Platform - Vendor Dependant Mode Specification - Programmer's Reference

CWA 14923-7:2004 replaces CWA 13937-7:2003 and should be read in conjunction with CWA 13937-7:2000, which contains the previous release of the J/XFS specification

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.

History

The main differences to the previous CWA13937-2000 are:

- JXFS_OC_ALM_ALARM renamed to JXFS_O_ALM_ALARM

1 Scope

This document describes the Alarm device classes based on the basic architecture of J/XFS which is similar to the JavaPOS architecture. It is event driven and asynchronous.

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

- Application
- Device Control and 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.

For Alarm Devices the basic Device Control class is extended with a method specific to this device which is described on the following pages..

2 Overview

The Alarm device is used to notify users, devices and other interested parties of security violations. The notification mechanism used by alarm devices to signal such events are device and manufacturer dependent.

3 Classes and Interfaces

The following classes and interfaces are used by the J/XFS Alarm Device Control.

Class or Interface	Name	Description	Extends or Implements
Interface	IjxfBaseControl	Base interface for all controls.	
Interface	IjxfAlarmControl	Base interface for all alarm controls. Contains method declarations specific to alarm devices.	Extends: IjxfBaseControl
Class	JxfAlarm	Class for alarm control.	Implements: IjxfAlarmControl

The following classes and interfaces are used by the J/XFS Alarm Device Service.

Class or Interface	Name	Description	Extends or Implements
Interface	IjxfBaseService	Base interface for all services.	
Interface	IjxfAlarmService	Base interface for all alarm services. Contains method declarations specific to alarm devices.	Extends: IjxfBaseService

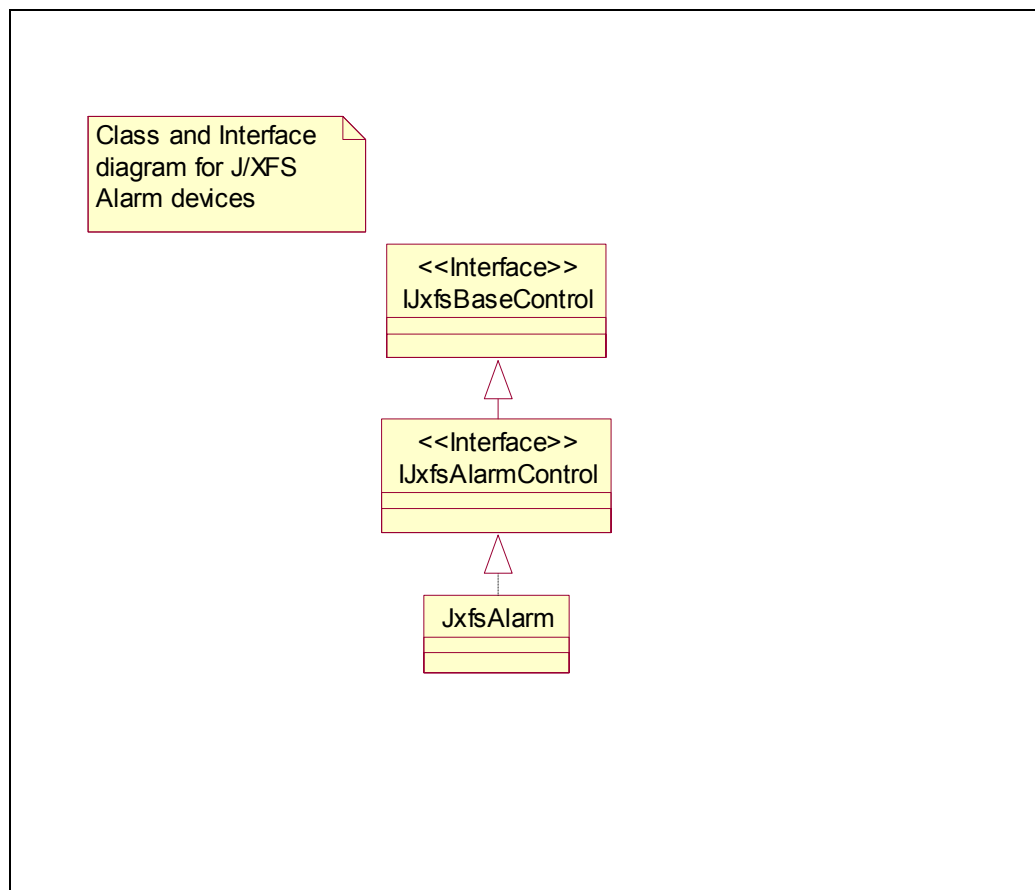
Remark on Device Services

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.

3.1 Class Diagram

The following class diagram shows the overall layout of the Alarm classes and interfaces provided by J/XFS.



3.2 Class and Interface Details

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

After processing has taken place, an *OutputComplete* – Event is generated which contains detailed information about the status of the operation, i.e. if it failed or succeeded, and eventually additional data as a result.

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

3.2.1 Access to properties

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

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

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

3.2.1.1 *getProperty*

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

3.2.1.2 *setProperty*

Syntax	Property <i>setProperty(void)</i> throws <i>JxfsException</i>;
Description	Sets the requested property.
Parameter	Single parameter of property type.
Event	No additional events are generated.
Exceptions	Some possible <i>JxfsException value codes</i> . See section on <i>JxfsExceptions</i> for other <i>JxfsException value codes</i> . JXFS_E_CLOSED JXFS_E_PARAMETER_INVALID JXFS_E_REMOTE JXFS_E_UNREGISTERED

3.2.2 Exceptions

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

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

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

3.3 IJxfsAlarmControl Summary

An alarm device is a device which is connected to the internal house alarm system. This method enables an "Application" to trigger this device. This would normally be done in the case of a bank robbery or an unauthorized access. The application can trigger this device even if an application "claim" some where else is pending.

This interface must be implemented by any device control that wants to use security violation signaling. If a device service has not implemented this interface, an exception is thrown.

Extends	Implements
IJxfsBaseControl	

Method	Return
alarm	void

3.4 JxfsAlarmControl Methods

Following method is specific to Alarm devices.

3.4.1 alarm

Syntax	identificationID alarm(boolean on) throws JxfsException;						
Remarks	Initiates a device alarm. This method is used to start or stop a notification when a security violation occurred.						
Notice	<i>This method will succeed, even when the device, which incorporates the alarm device, is in the state CLAIMED.</i>						
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;">Description</th> </tr> </thead> <tbody> <tr> <td><i>boolean</i></td> <td>on</td> <td>Turns the alarm signaling mechanism on when <i>true</i>, otherwise turns it off.</td> </tr> </tbody> </table>	Type	Name	Description	<i>boolean</i>	on	Turns the alarm signaling mechanism on when <i>true</i> , otherwise turns it off.
Type	Name	Description					
<i>boolean</i>	on	Turns the alarm signaling mechanism on when <i>true</i> , otherwise turns it off.					
Events	Events, which can be generated by this method.						

JxfsOperationCompleteEvent

When an *alarm* operation is completed a *JxfsOperationCompleteEvent* is sent to all registered listeners with following data:

Field	Value
<i>operationID</i>	JXFS_O_ALM_ALARM
<i>identificationID</i>	identificationID returned by method.
<i>result</i>	Common or device dependend error code. (See section on Error codes).
<i>data</i>	none

4 Support Classes

No support classes are used by this device.

5 Status Event Classes

There exists no device specific Status Event Classes.

6 Codes

6.1 Operation Codes

Following code specifies the method, which generated a *JxfsOperationCompleteEvent*.

6.1.1 JxfsAlarmControl

Value	Method
JXFS_O_ALM_ALARM	<i>alarm</i>

6.2 Exception Codes

Following table specifies exception codes that might occur as result of a method call. Exception codes are delivered to a caller during a method call.

The exception codes are already defined in the J/XFS-Base-Architecture and repeated here for easy access only.

6.2.1 Exception Code Summary and Description

Value	Description
JXFS_E_CLOSED	Device has not been opened yet.
JXFS_E_FAILURE	The operation failed.
JXFS_E_ILLEGAL	Illegal request.
JXFS_E_IO	Errors during IO-operation.
JXFS_E_NO_HARDWARE	No Device is connected to the workstation.
JXFS_E_NOT_CLAIMED	Device is not claimed by caller.
JXFS_E_NOT_SUPPORTED	Operation is not supported by device.
JXFS_E_OFFLINE	Device is offline.
JXFS_E_PARAMETER_INVALID	An invalid parameter was given to the operation.
JXFS_E_REMOTE	Communication error during remote call.
JXFS_E_TIMEOUT	A timeout has occurred.

6.2.2 IJxfsAlarmControl Exception Codes

Method	
alarm	
Exception Codes	
JXFS E CLOSED	x
JXFS E FAILURE	x
JXFS E ILLEGAL	x
JXFS E IO	x
JXFS E NO HARDWARE	x
JXFS E NOT SUPPORTED	x
JXFS E OFFLINE	x
JXFS E PARAMETER_INVALID	x
JXFS E REMOTE	x
JXFS E TIMEOUT	x

6.3 Error Codes

Following table specifies error codes that might occur as result of a method call. Error codes are delivered to a caller in field *result* of a *JxfsOperationCompleteEvent*.

6.3.1 Common Codes for all operations

Following codes can always occur as *result* of a *JxfsOperationCompleteEvent*:

Value	Description
JXFS_RC_SUCCESSFUL	Operation completed without error.
JXFS_E_TIMEOUT	A timeout during method execution occurred.

6.3.2 Error Code Summary and Description

Value	Description
JXFS_E_ALM_NOT_SUPPORTED	The device does not support alarm signaling.
JXFS_E_ALM_DEVICE_ERROR	An internal device error occurred.

Method	
alarm	
Error Codes	
JXFS_E_ALM_NOT_SUPPORTED	x
JXFS_E_ALM_DEVICE_ERROR	x

7 Constants

There are no constants defined for Alarm Devices.

7.1 Handling of *null* parameters

N/A

7.2 Handling of *null* return values

N/A

8 APPENDIX A : CEN/ISSS WORKSHOP 14923:2004 CORE MEMBERS :

DELARUE

DIEBOLD



DYNASTY



IBM



KAL

KEBA

LUTZ WOLF GRUPPE



NCR



NEXUS

SEIKO EPSON CORPORATION

WINCOR - NIXDORF

