

**CEN**

**CWA 16008-7**

**WORKSHOP**

August 2009

**AGREEMENT**

---

ICS 35.240.40

English version

**J/eXtensions for Financial Services (J/XFS) for the Java  
Platform - Release 2009 - Part 7: Alarm Device Class Interface -  
Programmer's Reference**

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

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

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

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

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



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

**Management Centre: Avenue Marnix 17, B-1000 Brussels**

---

© 2009 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.:CWA 16008-7:2009 E

## Contents

<b>Foreword</b> .....	<b>3</b>
<b>1 History</b> .....	<b>5</b>
<b>2 Scope</b> .....	<b>6</b>
<b>3 Overview</b> .....	<b>7</b>
<b>4 Classes and Interfaces</b> .....	<b>8</b>
<b>4.1 Class Diagram</b> .....	<b>9</b>
<b>4.2 Class and Interface Details</b> .....	<b>10</b>
4.2.1 Access to properties .....	10
<b>4.3 IJxfsAlarmControl</b> .....	<b>11</b>
4.3.1 Summary .....	11
4.3.2 Properties .....	11
4.3.3 Methods .....	11
<b>5 Support Classes</b> .....	<b>13</b>
<b>5.1 Summary</b> .....	<b>13</b>
<b>5.2 JxfsAlarmStatus</b> .....	<b>13</b>
5.2.1 Introduction .....	13
5.2.2 Summary .....	13
5.2.3 Properties .....	13
5.2.4 Constructors .....	13
<b>5.3 Enum Classes</b> .....	<b>14</b>
5.3.1 JxfsAlarmStatusEnum .....	14
5.3.2 JxfsALMStatusSelectorEnum .....	14
<b>6 Status Event Classes</b> .....	<b>15</b>
<b>6.1 Status Event Code Summary and Description</b> .....	<b>15</b>
<b>6.2 Status Event Details</b> .....	<b>15</b>
<b>7 Codes</b> .....	<b>16</b>
<b>7.1 Operation Codes</b> .....	<b>16</b>
<b>7.2 Error Codes</b> .....	<b>16</b>
7.2.1 Error Code Summary and Description .....	16
<b>8 Constants</b> .....	<b>17</b>
<b>8.1 Handling of <i>null</i> parameters</b> .....	<b>17</b>
<b>8.2 Handling of <i>null</i> return values</b> .....	<b>17</b>

## 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 J/XFS Workshop. J/XFS provides an API for Java applications which need to access financial devices. It is hardware independent and, by using 100% pure Java, also operating system independent.

The CEN J/XFS Workshop gathers suppliers (among others the J/XFS Forum members), service providers as well as banks and other financial service companies. A list of companies participating in this Workshop and in support of this CWA is available from the CEN Secretariat, and at [http://www.cen.eu/cenorm/sectors/sectors/iss/activity/jxfs\\_membership.asp](http://www.cen.eu/cenorm/sectors/sectors/iss/activity/jxfs_membership.asp). The specification was agreed upon by the J/XFS Workshop Meeting of 2009-05-6/9 in Brussels, and the final version was sent to CEN for publication on 2009-06-12.

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

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

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

This CWA is composed of the following parts:

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

## **CWA 16008-7:2009 (E)**

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://www.sun.com>. All other trademarks are trademarks of their respective owners.

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

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

## 1 History

Main differences to CWA 14923:2004 are:

- Numerical values for constants specified .
- Property *alarmStatus* added to the *IJxfAlarmControl* interface.
- New support classes: *JxfAlarmStatus*, *JxfAlarmStatusEnum* and *JxfALMStatusSelectorEnum*.
- New status event: *JXFS\_S\_ALM\_ALARM*.

Main differences to CWA 13937:2000 are:

- *JXFS\_OC\_ALM\_ALARM* renamed to *JXFS\_O\_ALM\_ALARM*

## 2 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.

### 3 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.

## 4 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	<i>IJxfsBaseControl</i>	Base interface for all controls.	
Interface	<i>IJxfsAlarmControl</i>	Base interface for all alarm controls. Contains method declarations specific to alarm devices.	Extends: <i>IJxfsBaseControl</i>
Class	<i>JxfsAlarm</i>	Class for alarm control.	Implements: <i>IJxfsAlarmControl</i>

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

Class or Interface	Name	Description	Extends or Implements
Interface	<i>IJxfsBaseService</i>	Base interface for all services.	
Interface	<i>IJxfsAlarmService</i>	Base interface for all alarm services. Contains method declarations specific to alarm devices.	Extends: <i>IJxfsBaseService</i>

### Remark on Device Services

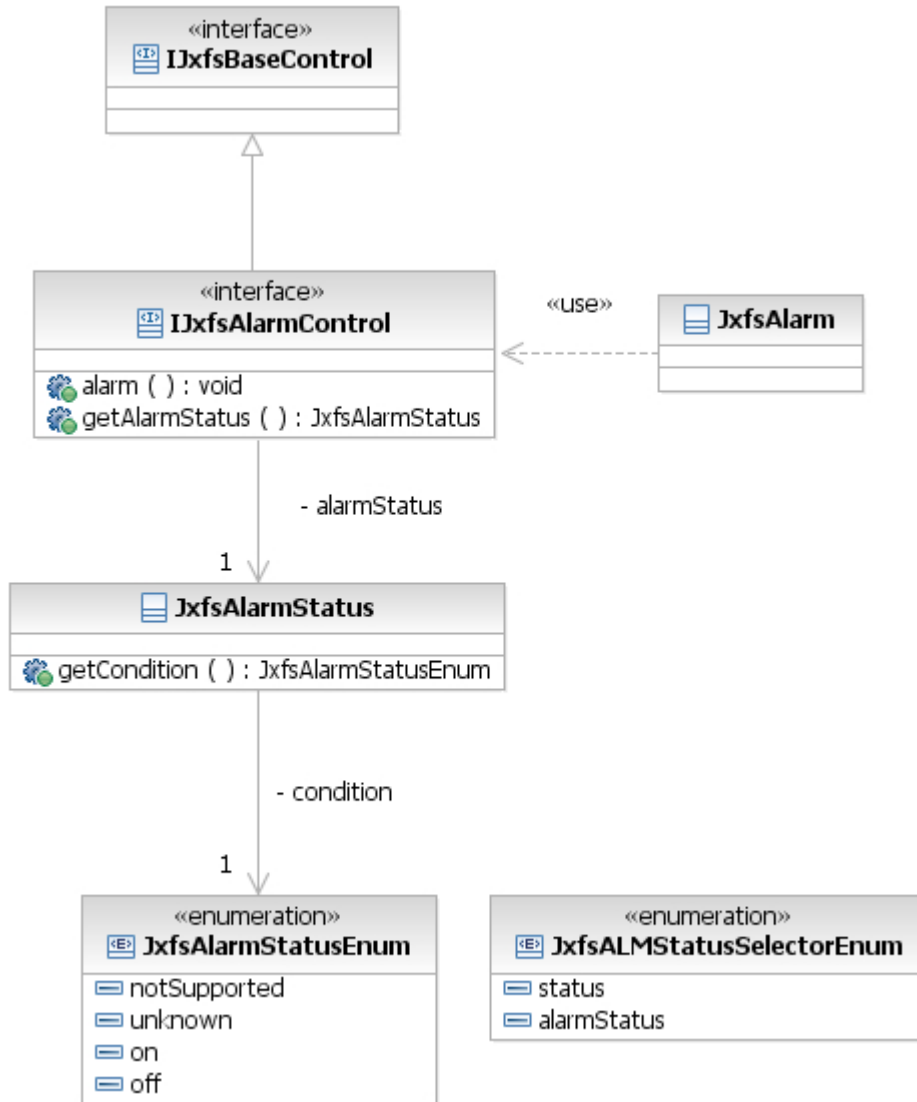
The Device Service interface is common for all device services of a specific type. It is used by the Device Controls to access the functionality of the device. This interface has to be implemented by any J/XFS Device Service.

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.



## 4.1 Class Diagram

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



## 4.2 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 Operation Complete Event is generated which contains detailed information about the status of the operation, i.e. if it failed or succeeded, and eventually additional data as a result.

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

### 4.2.1 Access to properties

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

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

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

#### 4.2.1.1 *getProperty*

<b>Syntax</b>	<b>Property <i>getProperty()</i> throws <i>JxfsException</i>;</b>
<b>Description</b>	Returns the requested property.
<b>Parameter</b>	<b>None</b>
<b>Event</b>	No additional events are generated.
<b>Exceptions</b>	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

#### 4.2.1.2 *setProperty*

<b>Syntax</b>	<b>void <i>setProperty( value)</i> throws <i>JxfsException</i>;</b>
<b>Description</b>	Sets the requested property.
<b>Parameter</b>	<b>Single parameter of property type.</b>
<b>Event</b>	No additional events are generated.
<b>Exceptions</b>	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

## 4.3 IJxfsAlarmControl

### 4.3.1 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 signalling. If a device service has not implemented this interface, an exception is thrown.

Extends	Implements
IJxfsBaseControl	

Property	Type	Access
<i>alarmStatus</i>	<i>JxfsAlarmStatus</i>	R

Method	Return
alarm	void

### 4.3.2 Properties

Following property is specific to Alarm devices.

#### 4.3.2.1 alarmStatus

<b>Type</b>	<i>JxfsAlarmStatus</i>
<b>Initial Value</b>	Depends on device type.
<b>Description</b>	Holds the current status for the alarm device. If <i>JxfsStatus.isOpen()</i> equals 'false', the returned object may not accurately reflect the status of the device.
<b>Events</b>	Status events will be generated when members of <i>JxfsAlarmStatus</i> object change. Refer to chapter 5.1 for more information.
<b>Exceptions</b>	No additional exceptions thrown.

### 4.3.3 Methods

Following method is specific to Alarm devices.

#### 4.3.3.1 alarm

<b>Syntax</b>	identificationID alarm( boolean on) throws JxfsException;		
<b>Remarks</b>	Initiates a device alarm. This method is used to start or stop a notification when a security violation occurred.		
<b>Notice</b>	<i>This method will succeed, even when the device, which incorporates the alarm device, is in the state CLAIMED.</i>		
<b>Parameter</b>	<b>Type</b>	<b>Name</b>	<b>Description</b>
	<i>boolean</i>	on	Turns the alarm signaling mechanism on when <i>true</i> , otherwise turns it off.
<b>Events</b>	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

## 5 Support Classes

### 5.1 Summary

Class	Description
<i>JxfsAlarmStatus</i>	Class for alarm specific status.
<i>JxfsAlarmStatusEnum</i>	Class to represent the possible alarm states.
<i>JxfsALMStatusSelectorEnum</i>	Class used for the base <code>getStatus(java.util.List)</code> method

### 5.2 JxfsAlarmStatus

#### 5.2.1 Introduction

All alarm specific status informations are kept in the `JxfsAlarmStatus` object, that can be queried by using the `getAlarmStatus()` method of the `JxfsAlarm` class.

#### 5.2.2 Summary

Extends	Implements
<code>JxfsType</code>	

Property	Type	Access
<code>condition</code>	<i>JxfsAlarmStatusEnum</i>	R

Constructor	Parameter
<code>JxfsAlarmStatus</code>	<code>condition</code>

Method	Return
<code>getProperty</code>	<i>Property</i>

#### 5.2.3 Properties

##### 5.2.3.1 condition

<b>Type</b>	<i>JxfsAlarmStatusEnum</i>
<b>Initial Value</b>	unknown
<b>Remarks</b>	Current status of the alarm.

#### 5.2.4 Constructors

##### 5.2.4.1 JxfsAlarmStatus

<b>Syntax</b>	<i><code>public JxfsAlarmStatus (JxfsAlarmStatusEnum condition) throws JxfsException</code></i>
<b>Exceptions</b>	Exceptions, which can be generated by this method. <code>JXFS_E_PARAMETER_INVALID</code> Generated if <i>condition</i> is a null reference.

### 5.3 Enum Classes

All enumerations are defined in terms of a class. The following describes all enumerated classes.

#### 5.3.1 JxfsAlarmStatusEnum

This enumerated data type represents the possible alarm states.

Extends	Implements
JxfsEnum	

Field	Description
notSupported	This DS does not support reporting the current alarm state.
unknown	The current alarm state is not known.
on	The alarm in the (external) system is active.
off	The alarm in the (external) system is not activ.

#### 5.3.2 JxfsALMStatusSelectorEnum

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

Extends	Implements
JxfsStatusSelectorEnum	

Field	Returned Type	Description
status	<i>JxfsStatus</i>	General status of the device
alarmStatus	<i>JxfsAlarmStatus</i>	Specifies if the device is in "alarm on" entry state or not

## 6 Status Event Classes

### 6.1 Status Event Code Summary and Description

Code	Value	Meaning
7010	JXFS_S_ALM_ALARM	the alarm on status has changed

### 6.2 Status Event Details

#### 5.2.1 JXFS\_S\_ALM\_ALARM

This event is sent whenever *JxfsAlarmStatus.condition* changes its state.

Field	Value
<i>Status</i>	JXFS_S_ALM_ALARM
<i>Details</i>	<i>JxfsAlarmStatus</i> object.

## 7 Codes

### 7.1 Operation Codes

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

Code	Value	Method
7003	JXFS_O_ALM_ALARM	<i>alarm</i>

### 7.2 Error Codes

Following table specifies error codes that might occur as result of an operation. Error codes are delivered to a caller either in field result of a *JxfsOperationCompleteEvent* or in a *JxfsException*.

#### 7.2.1 Error Code Summary and Description

Code	Value	Description
7001	JXFS_E_ALM_NOT_SUPPORTED	The device does not support alarm signaling.
7002	JXFS_E_ALM_DEVICE_ERROR	An internal device error occurred.

Method		
<i>alarm</i>		
Error Codes		
JXFS_E_ALM_NOT_SUPPORTED		x
JXFS_E_ALM_DEVICE_ERROR		x



## 8 Constants

There are no constants defined for Alarm Devices.

### 8.1 Handling of *null* parameters

N/A

### 8.2 Handling of *null* return values

N/A