



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

---

# WORKSHOP AGREEMENT

**CWA 13937-7**

August 2000

---

ICS 35.240.40

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part  
7: Alarm Device 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 13937-7:2000 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/worshop/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](mailto:iss@cenorm.be)). To submit questions and comments for the J/XFS specifications, please contact the CEN/ISSS Secretariat ([iss@cenorm.be](mailto: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](http://java.sun.com/nav/business/trademark_guidelines.html). All other trademarks are trademarks of their respective owners.

## Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>SCOPE.....</b>                        | <b>4</b>  |
| <b>2</b> | <b>OVERVIEW .....</b>                    | <b>5</b>  |
| <b>3</b> | <b>CLASS DIAGRAM .....</b>               | <b>6</b>  |
| <b>4</b> | <b>CLASS AND INTERFACE SUMMARY .....</b> | <b>7</b>  |
| <b>5</b> | <b>CLASS AND INTERFACE DETAILS .....</b> | <b>8</b>  |
| 5.1      | EXCEPTIONS .....                         | 8         |
| 5.2      | IJXFSALARMCONTROL.....                   | 9         |
| 5.2.2    | Methods.....                             | 9         |
| <b>6</b> | <b>SUPPORT CLASSES.....</b>              | <b>10</b> |
| <b>7</b> | <b>STATUS EVENT CLASSES .....</b>        | <b>11</b> |
| <b>8</b> | <b>CODES .....</b>                       | <b>12</b> |
| 8.1      | ERROR CODES .....                        | 12        |
| 8.2      | EXCEPTION CODES .....                    | 13        |
| 8.3      | STATUS CODES .....                       | 14        |
| 8.4      | OPERATION ID CODES.....                  | 15        |

## 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 alarm devices the basic Device Control structure is extended with various properties and methods specific to this device which are described on the following pages.