CEN

WORKSHOP

AGREEMENT

CWA 16008-10

August 2009

ICS 35.240.40

English version

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Release 2009 - Part 10: Check Reader/Scanner 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 16008-10:2009 E

Contents

This document is a preview generated by EVS

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/isss/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/isss/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).

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 - Release 2009 - 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)

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 document is a preview generated by EVS

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.

This document is a preview generated by EVS

History

Main Differences to CWA 14394-10:2004 are:

- open job handling clarified at base architecture level so specific chapter in this document is removed.
- specific declaration of result codes used by each job has been removed, and now result refers to common section at the end of the document.
- New JxfsCHKStatusSelectorEnum enumeration introduced to allow use of new getStatus method defined in base architecture documentation.

This document is a preview generated by EVS

1 Scope

This document describes the Check Reader/Scanner class based on the basic architecture of J/XFS which is similar to the JavaPOS architecture. It is event driven and asynchronous.

This specification has been superseded by the new part 13: Scanner Class Interface and it is now deprecated. It is strongly suggested to use the new device class interface for new implementations.

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

This document is a preview generated by EVS

2 Overview

2.1 Description

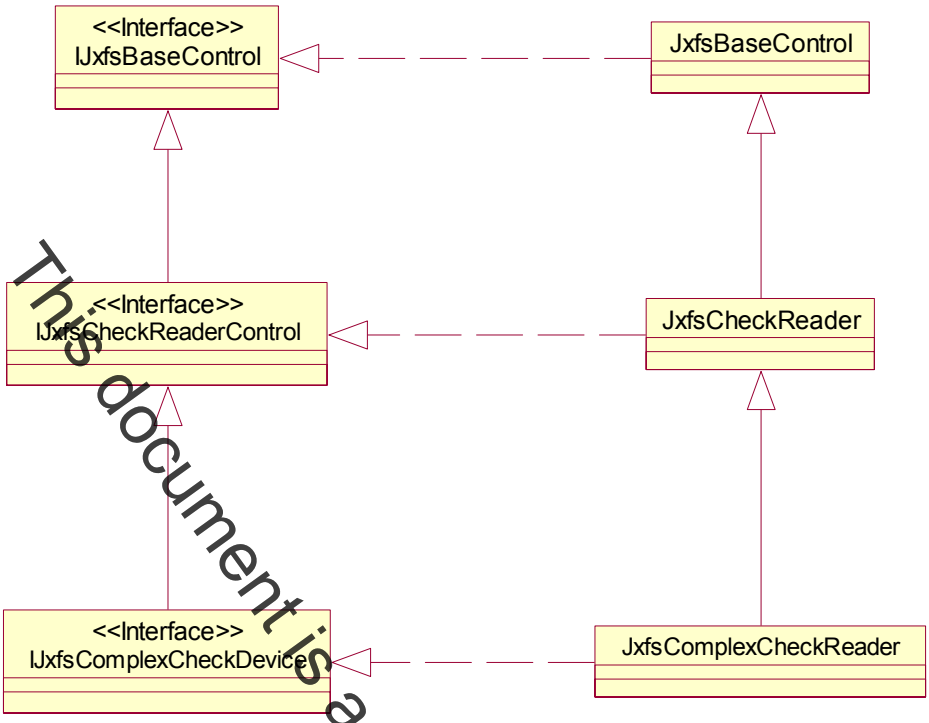
The J/XFS Check Reader/Scanner Device Support allows for the operation of devices with a range of features, from small hand-held read-only devices where checks are manually swiped through one at a time, to much larger devices which automatically feed checks by the batch past a reader, an encoder, an endorser, an optional image scanner, to be sorted into one of several pockets.

In the U.S. checks are always encoded in magnetic ink for reading by Magnetic Ink Recognition (MICR), and a single font is always used. In other areas some countries use MICR and some use Optical Character Recognition (OCR) character sets, with different fonts.

As well as the rest of J/XFS device controls, J/XFS Check Reader/Scanner devices use the event driven model and the same behavioral model. Therefore, the application will instantiate a J/XFS Check Reader/Scanner Device Control Object and then use the available methods to do I/O. When an I/O method is called, the J/XFS Check Reader/Scanner Device Service will attempt to process the requested I/O. If the request is invalid or an exception is encountered, the application will be notified by a J/XFS exception. Completion of the request will be reported by an event. Thus the application must register itself with the J/XFS Check Reader/Scanner Device Control Object for the various types of events it wishes to handle.

This document is a preview generated by EVS

2.2 Class Hierarchy



This document is a preview generated by EVS

2.3 Classes and Interfaces

The following classes and interfaces are used by the J/XFS CheckReader Device Controls. In order to support the definition of the different properties of the different devices (see Introduction), the Device Controls are defined in a class hierarchy.

Class or Interface	Name	Description	Extends or Implements
Interface	**IJxfsBaseControl**	Base interface for all the device controls. Contains methods common to all the device controls.	
Interface	**IJxfsCheckReaderControl**	Base interface for CheckReader controls. Contains method declarations specific to CheckReader controls.	Extends: **IJxfsBaseControl**
Interface	**IJxfsCheckReaderService**	Base interface for CheckReader services. Contains the methods specific to the device services for the CheckReader device category.	Extends: IJxfsBaseService
Interface	**IJxfsComplexCheckDevice**	Interface for complex check devices. Contains method declarations specific to complex check devices.	Extends: **IJxfsCheckReaderControl**
Interface	**IJxfsComplexCheckReaderService**	Interface for complex CheckReader services. Contains the methods specific to the device services for the complex check devices.	Extends: **IJxfsCheckReaderService**
Class	**JxfsBaseControl**	Base class for all the device controls. Contains properties common to all the device controls.	
Class	**JxfsCheckReader**	Base class for CheckReader controls. Contains properties specific to CheckReader device controls.	Extends: **JxfsBaseControl** Implements: **IJxfsCheckReaderControl**
Class	**JxfsComplexCheckReader**	Base class for check reader controls supporting the IJxfsComplexCheckDevice interface	Extends: **JxfsCheckReader** Implements: **IJxfsComplexCheckDevice**

This document is a preview generated by EVS