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

WORKSHOP AGREEMENT

CWA 13937-1

August 2000

ICS 35.240.40

J/eXtensions for Financial Services (J/XFS) for the Java Platform - Part 1: Base Architecture - Programmer's Reference

Ref. No CWA 13937-1:2000 E

Rue de Stassart, 36 • B-1050 Bruxelles
Tel : +32 2 550 08 11 • Fax : +32 2 550 08 19

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/isss/workshop/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 (isss@cenorm.be). To submit questions and comments for the J/XFS specifications, please contact the CEN/ISSS Secretariat (isss@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

1.1 Overview

J/XFS defines a standardized interface to all common financial devices which can be used by *applications and applets*[1] written in the Java programming language. One of the reasons why these new banking applications are written in the Java language is that these programs are supposed to run on many different hardware platforms. One of the main obstacles in doing platform independent programming is accessing devices.
One of the main goals of this standard is to allow access to banking devices in a 100% pure Java way on both thin and thick clients, e.g. on a network computer as well as in a Linux, WinNT, OS/2 or Unix workstation.

Another goal is to allow the remote access to devices on different machines. Additional efforts have to be done to find and access these devices. This is the main reason why central administration processes and an additional communication layer are also defined by this architecture.
If only local access to devices is needed, an implementation may omit this communication layer. No change is required to the Device Controls or Device Services. So, neither the application programmer nor the hardware manufacturer who programs a Device Service need be aware of whether or not a communication layer exists in the middle.

Due to the nature of network computers which are supported as clients, it is not possible to guarantee that local persistent storage possibilities exist on each client. Therefore, any configuration information must be kept on a central server. If local storage exists and no central configuration possibilities are required, all configuration information can also be kept on the local workstation.

The basic architecture of J/XFS is similar to the JavaPOS[2] 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 or applet
- Device Control and Manager
- Device Communication
- Device Service

The Device Control API defines the way a Java application or applet can communicate with a specific device. Additionally, the Device Control layer contains the central Device Manager which organizes access and location of the services. It is the central coordinating instance in any Java VM which must access financial devices.
The Device Communication Layer is the layer which resolves the sharing of devices. It is invisible to the application. The only exception is that network errors are presented to the application. It must be able to cope with lost connections.
The Device Service is the layer supplied by the device manufacturer for use with J/XFS. It has a defined API which allows the Device Control and Device Communication layer to request device actions and translates them into the device specific commands which are then sent to the physical attached device. The way of connecting to the local device is not defined in this standard, it is rather left to the service provider. In the case of devices which attach through the serial or parallel ports the Java CommAPI may be used. Thus, the Device Service layer may not be 100% pure Java but the complete basic infrastructure of J/XFS is.

[1] J/XFS is designed to be also usable by applets in a browser e.g. on a network computer. So, for the remainder of the document, ‘application‘ also always means ‘applet’.
[2] JavaPOS (Java point of sale) is an initiative for the retail industry with the goal of providing unified device access to POS devices. See http://www.javapos.com .