

**CEN**

**CWA 15748-7**

**WORKSHOP**

July 2008

**AGREEMENT**

---

ICS 35.240.50

Supersedes CWA 15748-7:2008, February

English version

**Extensions for Financial Services (XFS) interface specification -  
Release 3.10 - Part 7: 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: rue de Stassart, 36 B-1050 Brussels**

## Table of Contents

---

|  |           |
|--|-----------|
| <b>Foreword</b> .....  | <b>4</b>  |
| <b>1. Introduction</b> .....   | <b>7</b>  |
| 1.1 Background to Release 3.10 .....                                   | 7         |
| 1.2 XFS Service-Specific Programming .....                             | 7         |
| <b>2. Check Readers and Scanners</b> .....                             | <b>8</b>  |
| <b>3. References</b> .....   | <b>9</b>  |
| <b>4. Info Commands</b> .....  | <b>10</b> |
| 4.1 WFS_INF_CHK_STATUS .....   | 10        |
| 4.2 WFS_INF_CHK_CAPABILITIES .....                                     | 13        |
| 4.3 WFS_INF_CHK_FORM_LIST .....  | 16        |
| 4.4 WFS_INF_CHK_MEDIA_LIST .....                                       | 17        |
| 4.5 WFS_INF_CHK_QUERY_FORM .....                                       | 18        |
| 4.6 WFS_INF_CHK_QUERY_MEDIA .....                                      | 20        |
| 4.7 WFS_INF_CHK_QUERY_FIELD .....                                      | 22        |
| <b>5. Execute Commands</b> .....                                       | <b>24</b> |
| 5.1 WFS_CMD_CHK_PROCESS_FORM .....                                     | 24        |
| 5.2 WFS_CMD_CHK_RESET .....  | 27        |
| 5.3 WFS_CMD_CHK_SET_GUIDANCE_LIGHT .....                               | 28        |
| 5.4 WFS_CMD_CHK_POWER_SAVE_CONTROL .....                               | 29        |
| <b>6. Events</b> .....   | <b>30</b> |
| 6.1 WFS_EXEE_CHK_NOMEDIA .....   | 30        |
| 6.2 WFS_EXEE_CHK_MEDIAINSERTED .....                                   | 31        |
| 6.3 WFS_SRVE_CHK_MEDIAINSERTED .....                                   | 32        |
| 6.4 WFS_EXEE_CHK_FIELDERROR .....                                      | 33        |
| 6.5 WFS_EXEE_CHK_FIELDWARNING .....                                    | 34        |
| 6.6 WFS_USRE_CHK_INKTHRESHOLD .....                                    | 35        |
| 6.7 WFS_SRVE_CHK_MEDIADETECTED .....                                   | 36        |
| 6.8 WFS_SRVE_CHK_DEVICEPOSITION .....                                  | 37        |
| 6.9 WFS_SRVE_CHK_POWER_SAVE_CHANGE .....                               | 38        |
| <b>7. Forms Language Usage</b> .....                                   | <b>39</b> |
| 7.1 Definition Syntax .....  | 40        |
| 7.2 XFS form/media definition files in multi-vendor environments ..... | 41        |
| 7.3 Form and Media Measurements .....                                  | 42        |
| 7.4 Form Definition .....  | 43        |
| 7.5 Field Definition .....   | 44        |

7.6 Media Definition ..... 47

8. C - Header file ..... 48

This document is a preview generated by EVS

## Foreword

---

This CWA is revision 3.10 of the XFS interface specification.

The CEN/ISSS XFS Workshop gathers suppliers 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.

This CWA was formally approved by the XFS Workshop meeting on 2007-11-29. The specification is continuously reviewed and commented in the CEN/ISSS Workshop on XFS. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this revision 3.10.

The CWA is published as a multi-part document, consisting of:

Part 1: Application Programming Interface (API) - Service Provider Interface (SPI) - Programmer's Reference

Part 2: Service Classes Definition - Programmer's Reference

Part 3: Printer and Scanning Device Class Interface - Programmer's Reference

Part 4: Identification Card Device Class Interface - Programmer's Reference

Part 5: Cash Dispenser Device Class Interface - Programmer's Reference

Part 6: PIN Keypad Device Class Interface - Programmer's Reference

Part 7: Check Reader/Scanner Device Class Interface - Programmer's Reference

Part 8: Depository Device Class Interface - Programmer's Reference

Part 9: Text Terminal Unit Device Class Interface - Programmer's Reference

Part 10: Sensors and Indicators Unit Device Class Interface - Programmer's Reference

Part 11: Vendor Dependent Mode Device Class Interface - Programmer's Reference

Part 12: Camera Device Class Interface - Programmer's Reference

Part 13: Alarm Device Class Interface - Programmer's Reference

Part 14: Card Embossing Unit Device Class Interface - Programmer's Reference

Part 15: Cash-In Module Device Class Interface - Programmer's Reference

Part 16: Card Dispenser Device Class Interface - Programmer's Reference

Part 17: Barcode Reader Device Class Interface - Programmer's Reference

Part 18: Item Processing Module Device Class Interface - Programmer's Reference

Parts 19 - 28: Reserved for future use.

Parts 29 through 47 constitute an optional addendum to this CWA. They define the integration between the SNMP standard and the set of status and statistical information exported by the Service Providers.

Part 29: XFS MIB Architecture and SNMP Extensions - Programmer's Reference

Part 30: XFS MIB Device Specific Definitions - Printer Device Class

Part 31: XFS MIB Device Specific Definitions - Identification Card Device Class

Part 32: XFS MIB Device Specific Definitions - Cash Dispenser Device Class

Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class

Part 34: XFS MIB Device Specific Definitions - Check Reader/Scanner Device Class

Part 35: XFS MIB Device Specific Definitions - Depository Device Class

Part 36: XFS MIB Device Specific Definitions - Text Terminal Unit Device Class

Part 37: XFS MIB Device Specific Definitions - Sensors and Indicators Unit Device Class

Part 38: XFS MIB Device Specific Definitions - Camera Device Class

Part 39: XFS MIB Device Specific Definitions - Alarm Device Class

Part 40: XFS MIB Device Specific Definitions - Card Embossing Unit Class

Part 41: XFS MIB Device Specific Definitions - Cash-In Module Device Class

Part 42: Reserved for future use.

Part 43: XFS MIB Device Specific Definitions - Vendor Dependent Mode Device Class

Part 44: XFS MIB Application Management

Part 45: XFS MIB Device Specific Definitions - Card Dispenser Device Class

Part 46: XFS MIB Device Specific Definitions - Barcode Reader Device Class

Part 47: XFS MIB Device Specific Definitions - Item Processing Module Device Class

Parts 48 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 62: Printer Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.02 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.03 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.01 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.02 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is available online from <http://www.cen.eu/iss/Workshop/XFS>.

The information in this document represents the Workshop's current views on the issues discussed as of the date of publication. It is furnished for informational purposes only and is subject to change without notice. CEN/ISSS makes no warranty, express or implied, with respect to this document.

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.

Revision History:

|      |                   |   |
|------|-------------------|---|
| 1.0  | May 24, 1993      | Initial release of API and SPI specification.   |
| 1.11 | February 3, 1995  | Separation of specification into separate documents for API/SPI and service class definitions; with updates.  |
| 3.0  | October 18, 2000  | Update release encompassing:<br>- Reintroduced with a command set targeted at stand alone check readers and scanners<br>- UNICODE support<br><br>For a detailed description see CHK migration document from version 2.0 to version 3.0, revision 1.00, October 18 <sup>th</sup> 2000. |
| 3.10 | November 29, 2007 | For a description of changes see CWA 15748-66:2007 CHK Migration from Version 3.0 (see CWA 14050) to Version 3.10.  |

This document is a preview generated by EVS

## 1. Introduction

---

### 1.1 Background to Release 3.10

---

The CEN/ISSS XFS Workshop aims to promote a clear and unambiguous specification defining a multi-vendor software interface to financial peripheral devices. The XFS (eXtensions for Financial Services) specifications are developed within the CEN/ISSS (European Committee for Standardization/Information Society Standardization System) Workshop environment. CEN/ISSS Workshops aim to arrive at a European consensus on an issue that can be published as a CEN Workshop Agreement (CWA).

The CEN/ISSS XFS Workshop encourages the participation of both banks and vendors in the deliberations required to create an industry standard. The CEN/ISSS XFS Workshop achieves its goals by focused sub-groups working electronically and meeting quarterly.

Release 3.10 of the XFS specification is based on a C API and is delivered with the continued promise for the protection of technical investment for existing applications. This release of the XFS specification has been prompted by a series of factors.

There has been a technical imperative to extend the scope of the existing specification to include new devices, such as the Barcode Reader, Card Dispenser and Item Processing Module.

Similarly, there has also been pressure, through implementation experience and additional requirements, to extend the functionality and capabilities of the existing devices covered by the specification.

### 1.2 XFS Service-Specific Programming

---

The service classes are defined by their service-specific commands and the associated data structures, error codes, messages, etc. These commands are used to request functions that are specific to one or more classes of Service Providers, but not all of them, and therefore are not included in the common API for basic or administration functions.

When a service-specific command is common among two or more classes of Service Providers, the syntax of the command is as similar as possible across all services, since a major objective of XFS is to standardize function codes and structures for the broadest variety of services. For example, using the **WFSExecute** function, the commands to read data from various services are as similar as possible to each other in their syntax and data structures.

In general, the specific command set for a service class is defined as a superset of the specific capabilities likely to be provided by the developers of the services of that class; thus any particular device will normally support only a subset of the defined command set.

There are three cases in which a Service Provider may receive a service-specific command that it does not support:

The requested capability is defined for the class of Service Providers by the XFS specification, the particular vendor implementation of that service does not support it, and the unsupported capability is *not* considered to be fundamental to the service. In this case, the Service Provider returns a successful completion, but does no operation. An example would be a request from an application to turn on a control indicator on a passbook printer; the Service Provider recognizes the command, but since the passbook printer it is managing does not include that indicator, the Service Provider does no operation and returns a successful completion to the application.

The requested capability is defined for the class of Service Providers by the XFS specification, the particular vendor implementation of that service does not support it, and the unsupported capability *is* considered to be fundamental to the service. In this case, a `WFS_ERR_UNSUPP_COMMAND` error is returned to the calling application. An example would be a request from an application to a cash dispenser to dispense coins; the Service Provider recognizes the command but, since the cash dispenser it is managing dispenses only notes, returns this error.

The requested capability is *not* defined for the class of Service Providers by the XFS specification. In this case, a `WFS_ERR_INVALID_COMMAND` error is returned to the calling application.

This design allows implementation of applications that can be used with a range of services that provide differing subsets of the functionalities that are defined for their service class. Applications may use the **WFSGetInfo** and **WFSAsyncGetInfo** commands to inquire about the capabilities of the service they are about to use, and modify their behavior accordingly, or they may use functions and then deal with `WFS_ERR_UNSUPP_COMMAND` error returns to make decisions as to how to use the service.

## 2. Check Readers and Scanners

---

This specification describes the XFS service class of check readers and scanners. Check image scanners are treated as a special case of check readers, i.e. image-enabled instances of the latter. This class includes devices with a range of features, from small hand-held read-only devices through which checks are manually swiped one at a time, to desktop units which automatically feed the check one at a time; recording the MICR data and check image, and endorse or encode the check. The specification of this service class includes definitions of the service-specific commands that can be issued, using the **WFSAsyncExecute**, **WFSExecute**, **WFSGetInfo** and **WFSAsyncGetInfo** functions.

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

In all countries, typical fields found encoded on a check include the bank ID number and the account number. Part of the processing done by the bank is to also encode the amount on the check, usually done by having an operator enter the handwritten or typewritten face amount on a numeric keypad.

This service class is currently defined only for attended branch service.

Document is a preview generated by EVS

### 3. References

---

- |   |
|---|
| 1. XFS Application Programming Interface (API)/Service Provider Interface ( SPI), Programmer's Reference<br>Revision 3.10 |
|---|

This document is a preview generated by EVS