



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

WORKSHOP AGREEMENT

CWA 14050-16

November 2000

ICS 35.200; 35.240.15; 35.240.40

Extensions for Financial Services (XFS) interface specification -
Release 3.0 - Part 16: Application Programming Interface (API) - Service
Provider Interface (SPI) - Migration from Version 2.0 (see CWA 13449) to
Version 3.0 (this CWA) - 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 14050-16:2000 E

Table of Contents

Foreword.....	3
1. General.....	5
2. New Chapters	5
2.1 Vendor Dependent Mode	5
2.2 References.....	5
3. Changes to existing Chapters.....	5
3.1 Configuration Information	5
3.2 Exclusive Service and Device Access	9
3.3 Lock Policy for Independant Devices	9
3.4 Notification Mechanisms — Registering for Events	10
4. Changes to Application Programming Interface (API) Functions.....	11
4.1 WFSExecute	11
4.2 WFSAsyncExecute.....	12
4.3 WFSGetInfo	13
4.4 WFSAsyncGetInfo.....	14
4.5 WFSOpen.....	15
4.6 WFSAsyncOpen	15
5. Changes to Service Provider Interface (SPI) Functions	15
5.1 WFPExecute	15
5.2 WFPGetInfo	17
6. Changes to Configuration Functions	18
6.1 WFMCreateKey.....	18
7. New Events	19
7.1 Lock Requested	19
8. Changes to existing Events	19
8.1 Device Status Changes	19
8.2 Hardware and Software Errors	20
9. New Error Codes	21
10. Changes to C – Header files.....	22
10.1 XFSAPI.H.....	22
10.2 XFSCONF.H	27

Foreword

This CWA is revision 3.0 of the XFS interface specification.

The move from an XFS 2.0 specification (CWA 13449) to a 3.0 specification has been prompted by a series of factors.

Initially, there has been a technical imperative to extend the scope of the existing specification of the XFS Manager to include new devices, such as the Card Embossing Unit.

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

Finally, it is also clear that our customers and the market are asking for an update to a specification, which is now over 2 years old. Increasing market acceptance and the need to meet this demand is driving the Workshop towards this release.

The clear direction of the CEN/ISSS XFS Workshop, therefore, is the delivery of a new Release 3.0 specification based on a C API. It will be delivered with the promise of the protection of technical investment for existing applications and the design to safeguard future developments.

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 2000-10-18. 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 3.0.

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 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 Class Interface - Programmer's Reference

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

Part 16: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 17: Printer Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 18: Identification Card Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 19: Cash Dispenser Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 20: PIN Keypad Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 21: Depository Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 22: Text Terminal Unit Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 23: Sensors and Indicators Unit Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 24: Camera Device Class Interface - Migration from Version 2.0 (see CWA 13449) to Version 3.0 (this CWA) - Programmer's Reference

Part 25: Identification Card Device Class Interface - PC/SC Integration Guidelines

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.cenorm.be/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.

1. General

In the Configuration Information chapter the reference to Windows 3.1 and Windows for Workgroups is removed. The configuration information is now partly moved to HKEY_LOCAL_MACHINE and partly to HKEY_USERS.

The hardware and software error events now return information about the physical service name and an action that is required to manage the error.

2. New Chapters

2.1 Vendor Dependent Mode

XFS compliant applications must comply with the following:

- Every XFS application should open a session with the VDM ServiceProvider passing a valid AppId and then register for all VDM entry and exit notices.
- Before opening any session with any other XFS Service Provider, check the status of the VDM Service Provider. If Vendor Dependent Mode is not "Inactive", do not open a session.
- When getting a VDM entry notice, close all open sessions with all other XFS Service Providers as soon as possible and issue an acknowledgement for the entry to VDM.
- When getting a VDM exit notice, acknowledge at once.
- When getting a VDM exited notice, re-open any required sessions with other XFS Service Providers.

This is mandatory for self-service but optional for branch.

2.2 References

1. XFS Service Classes Definition, Programmer's Reference Revision 3.00, October 18, 2000

3. Changes to existing Chapters

3.1 Configuration Information

The XFS Manager uses its configuration information to define the relationships among the applications and the service providers. In particular, this information defines the mapping between the logical service interface presented at the API (via logical service name) and the appropriate service provider entry points.

The configuration information also includes specific information about logical services and service providers, some of which is common to all solution providers; it may also include information about physical services, if any are present on the system, and vendor-specific information. The location of the information is transparent to both applications and service providers; they always store and retrieve it using the configuration functions provided by the XFS Manager, as described in Section 7, for portability across Windows platforms.

It is the responsibility of solution providers, and the developers of each service provider, to implement the appropriate setup and management utilities, to create and manage the configuration information about the XFS subsystem configuration and its service providers, using the configuration functions.

These functions are used by service providers and applications to write and retrieve the configuration information for a XFS subsystem, which is stored in a hierarchical structure called the Windows Registry. The structure and the functions are based on the Win32 Registry architecture and API functions, and are implemented in Windows NT/98 and future versions of Windows using the Registry and the associated functions.