



SLOVENSKI STANDARD SIST CWA 16926-65:2023

01-april-2023

Specifikacija vmesnika razširitev za finančne storitve (XFS), izdaja 3.50 - 65. del: Vmesnik razreda naprave s tipkovnico PIN - Referenca za programerje - Prehod z različice 3.40 (CWA 16926:2020) na različico 3.50 (ta CWA)

Extensions for Financial Services (XFS) interface specification Release 3.50 - Part 65: PIN Keypad Device Class Interface - Programmer's Reference - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA)

standards.iteh.ai

[SIST CWA 16926-65:2023](https://standards.iteh.ai/catalog/standards/sist/21fd83f4-55bd-4c7f-88e2-8fb2ab88dc1a/sist-cwa-16926-65-2023)

<https://standards.iteh.ai/catalog/standards/sist/21fd83f4-55bd-4c7f-88e2-8fb2ab88dc1a/sist-cwa-16926-65-2023>

Ta slovenski standard je istoveten z: CWA 16926-65:2023

ICS:

35.200	Vmesniška in povezovalna oprema	Interface and interconnection equipment
35.240.15	Identifikacijske kartice. Čipne kartice. Biometrija	Identification cards. Chip cards. Biometrics
35.240.40	Uporabniške rešitve IT v bančništvu	IT applications in banking

SIST CWA 16926-65:2023

en,fr,de

CEN**CWA 16926-65****WORKSHOP**

January 2023

AGREEMENT

ICS 35.200; 35.240.15; 35.240.40

English version

**Extensions for Financial Services (XFS) interface
specification Release 3.50 - Part 65: PIN Keypad Device
Class Interface - Programmer's Reference - Migration from
Version 3.40 (CWA 16926:2020) to Version 3.50 (this
CWA)**

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-CENELEC 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, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

© 2023 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.:CWA 16926-65:2023 E

Table of Contents

European Foreword	6
1. Introduction	10
1.1 Background to Release 3.50	10
1.2 XFS Service-Specific Programming	10
2. PIN Keypad	11
2.1 Encrypting Touch Screen (ETS).....	13
3. References	16
4. Info Commands	18
4.1 WFS_INF_PIN_STATUS.....	18
4.2 WFS_INF_PIN_CAPABILITIES	22
4.3 WFS_INF_PIN_KEY_DETAIL.....	43
4.4 WFS_INF_PIN_FUNCKEY_DETAIL.....	45
4.5 WFS_INF_PIN_HSM_TDATA	48
4.6 WFS_INF_PIN_KEY_DETAIL_EX.....	49
4.7 WFS_INF_PIN_SECUREKEY_DETAIL.....	52
4.8 WFS_INF_PIN_QUERY_LOGICAL_HSM_DETAIL.....	56
4.9 WFS_INF_PIN_QUERY_PCIPTS_DEVICE_ID	57
4.10 WFS_INF_PIN_GET_LAYOUT.....	58
4.11 WFS_INF_PIN_KEY_DETAIL_340.....	62
5. Execute Commands	64
5.1 Normal PIN Commands	64
5.1.1 WFS_CMD_PIN_CRYPT	64
5.1.2 WFS_CMD_PIN_IMPORT_KEY	67
5.1.3 WFS_CMD_PIN_DERIVE_KEY	70
5.1.4 WFS_CMD_PIN_GET_PIN.....	72
5.1.5 WFS_CMD_PIN_LOCAL_DES	75
5.1.6 WFS_CMD_PIN_CREATE_OFFSET	77
5.1.7 WFS_CMD_PIN_LOCAL_EUROCHEQUE.....	79
5.1.8 WFS_CMD_PIN_LOCAL_VISA.....	81
5.1.9 WFS_CMD_PIN_PRESENT_IDC.....	83
5.1.10 WFS_CMD_PIN_GET_PINBLOCK	85
5.1.11 WFS_CMD_PIN_GET_DATA	87
5.1.12 WFS_CMD_PIN_INITIALIZATION	90
5.1.13 WFS_CMD_PIN_LOCAL_BANKSYS	92
5.1.14 WFS_CMD_PIN_BANKSYS_IO	93
5.1.15 WFS_CMD_PIN_RESET.....	94
5.1.16 WFS_CMD_PIN_HSM_SET_TDATA.....	95
5.1.17 WFS_CMD_PIN_SECURE_MSG_SEND.....	97
5.1.18 WFS_CMD_PIN_SECURE_MSG_RECEIVE	99
5.1.19 WFS_CMD_PIN_GET_JOURNAL	101
5.1.20 WFS_CMD_PIN_IMPORT_KEY_EX.....	102
5.1.21 WFS_CMD_PIN_ENC_IO.....	105
5.1.22 WFS_CMD_PIN_HSM_INIT.....	107
5.1.23 WFS_CMD_PIN_SECUREKEY_ENTRY	108

5.1.24	WFS_CMD_PIN_GENERATE_KCV	111
5.1.25	WFS_CMD_PIN_SET_GUIDANCE_LIGHT	112
5.1.26	WFS_CMD_PIN_MAINTAIN_PIN.....	114
5.1.27	WFS_CMD_PIN_KEYPRESS_BEEP	115
5.1.28	WFS_CMD_PIN_SET_PINBLOCK_DATA	116
5.1.29	WFS_CMD_PIN_SET_LOGICAL_HSM.....	117
5.1.30	WFS_CMD_PIN_IMPORT_KEYBLOCK	118
5.1.31	WFS_CMD_PIN_POWER_SAVE_CONTROL	119
5.1.32	WFS_CMD_PIN_DEFINE_LAYOUT.....	120
5.1.33	WFS_CMD_PIN_START_AUTHENTICATE.....	121
5.1.34	WFS_CMD_PIN_AUTHENTICATE.....	123
5.1.35	WFS_CMD_PIN_GET_PINBLOCK_EX	126
5.1.36	WFS_CMD_PIN_SYNCHRONIZE_COMMAND.....	128
5.1.37	WFS_CMD_PIN_CRYPT_340	129
5.1.38	WFS_CMD_PIN_GET_PINBLOCK_340.....	133
5.1.39	WFS_CMD_PIN_IMPORT_KEY_340.....	135
5.2	Common commands for Remote Key Loading Schemes.....	138
5.2.1	WFS_CMD_PIN_START_KEY_EXCHANGE.....	138
5.3	Remote Key Loading Using Signatures	139
5.3.1	WFS_CMD_PIN_IMPORT_RSA_PUBLIC_KEY	139
5.3.2	WFS_CMD_PIN_EXPORT_RSA_ISSUER_SIGNED_ITEM.....	142
5.3.3	WFS_CMD_PIN_IMPORT_RSA_SIGNED_DES_KEY	144
5.3.4	WFS_CMD_PIN_GENERATE_RSA_KEY_PAIR	147
5.3.5	WFS_CMD_PIN_EXPORT_RSA_EPP_SIGNED_ITEM.....	149
5.4	Remote Key Loading with Certificates	151
5.4.1	WFS_CMD_PIN_LOAD_CERTIFICATE.....	151
5.4.2	WFS_CMD_PIN_GET_CERTIFICATE.....	152
5.4.3	WFS_CMD_PIN_REPLACE_CERTIFICATE.....	153
5.4.4	WFS_CMD_PIN_IMPORT_RSA_ENCIPHERED_PKCS7_KEY.....	154
5.4.5	WFS_CMD_PIN_LOAD_CERTIFICATE_EX	156
5.4.6	WFS_CMD_PIN_IMPORT_RSA_ENCIPHERED_PKCS7_KEY_EX	158
5.5	EMV	162
5.5.1	WFS_CMD_PIN_EMV_IMPORT_PUBLIC_KEY	162
5.5.2	WFS_CMD_PIN_DIGEST	165
5.6	Entering and Changing a Password	166
5.6.1	WFS_CMD_PIN_PASSWORD_ENTRY	166
6.	Events.....	170
6.1	WFS_EXEE_PIN_KEY	170
6.2	WFS_SRVE_PIN_INITIALIZED	171
6.3	WFS_SRVE_PIN_ILLEGAL_KEY_ACCESS	172
6.4	WFS_SRVE_PIN_OPT_REQUIRED.....	173
6.5	WFS_SRVE_PIN_CERTIFICATE_CHANGE.....	174
6.6	WFS_SRVE_PIN_HSM_TDATA_CHANGED.....	175
6.7	WFS_SRVE_PIN_HSM_CHANGED.....	176
6.8	WFS_EXEE_PIN_ENTERDATA.....	177
6.9	WFS_SRVE_PIN_DEVICEPOSITION.....	178
6.10	WFS_SRVE_PIN_POWER_SAVE_CHANGE	179
6.11	WFS_EXEE_PIN_LAYOUT.....	180
6.12	WFS_EXEE_PIN_DUKPT_KSN	181
6.13	WFS_SRVE_PIN_PASSWORD_CLEARED	182

7. C - Header File	183
8. Appendix-A	206
8.1 Remote Key Loading Using Signatures	207
8.1.1 RSA Data Authentication and Digital Signatures	207
8.1.2 RSA Secure Key Exchange using Digital Signatures	208
8.1.3 Initialization Phase – Signature Issuer and ATM PIN	210
8.1.4 Initialization Phase – Signature Issuer and Host	211
8.1.5 Key Exchange – Host and ATM PIN	212
8.1.6 Key Exchange (with random number) – Host and ATM PIN	213
8.1.7 Enhanced RKL, Key Exchange (with random number) – Host and ATM PIN	216
8.1.8 Default Keys and Security Item loaded during manufacture.....	217
8.2 Remote Key Loading Using Certificates	218
8.2.1 Certificate Exchange and Authentication.....	218
8.2.2 Remote Key Exchange.....	220
8.2.3 Replace Certificate.....	221
8.2.4 Primary and Secondary Certificates.....	222
8.2.5 TR34 BIND To Host.....	223
8.2.6 TR34 Key Transport.....	224
8.2.7 TR34 REBIND To New Host	227
8.2.8 TR34 Force REBIND To New Host	228
8.2.9 TR34 UNBIND From Host	229
8.2.10 TR34 Force UNBIND From Host.....	230
8.3 German ZKA GeldKarte (Deutsche Kreditwirtschaft).....	231
8.3.1 How to use the SECURE_MSG commands.....	231
8.3.2 Protocol WFS_PIN_PROTISOAS	232
8.3.3 Protocol WFS_PIN_PROTISOLZ	233
8.3.4 Protocol WFS_PIN_PROTISOPS.....	234
8.3.5 Protocol WFS_PIN_PROTCHIPZKA	235
8.3.6 Protocol WFS_PIN_PROTRAWDATA.....	236
8.3.7 Protocol WFS_PIN_PROTPBM.....	237
8.3.8 Protocol WFS_PIN_PROTHSMLDI	238
8.3.9 Protocol WFS_PIN_PROTGENAS	239
8.3.10 Protocol WFS_PIN_PROTCHIPINCHG.....	243
8.3.11 Protocol WFS_PIN_PROTPINCOMP.....	244
8.3.12 Protocol WFS_PIN_PROTISOPINCHG	246
8.3.13 Command Sequence.....	247
8.4 EMV Support.....	254
8.4.1 Keys loading.....	254
8.4.2 PIN Block Management	256
8.4.3 SHA-1 Digest.....	257
8.5 French Cartes Bancaires.....	258
8.5.1 Data Structure for WFS_CMD_PIN_ENC_IO	258
8.5.2 Command Sequence.....	260
8.6 Secure Key Entry	262
8.6.1 Keyboard Layout.....	262
8.6.2 Command Usage - WFS_CMD_PIN_SECUREKEY_DETAIL and WFS_CMD_PIN_IMPORT_KEY.....	266
8.6.3 Command Usage - WFS_INF_PIN_GET_LAYOUT and WFS_CMD_PIN_IMPORT_KEY_340.....	267
8.7 WFS_PIN_USERRESTRICTEDKEYENCKEY key usage.....	268
8.7.1 Command Usage	268
8.8 WFS_CMD_PIN_IMPORT_KEY_340 command Input/Output Parameters.....	273
8.8.1 Importing a 3DES 16-byte terminal master key using signature-based remote key loading (SRKL): 274	
8.8.2 Importing a 16-byte DES key for PIN encryption with a key check value in the input	276
8.8.3 Importing a 16-byte DES key for MACing (MAC Algorithm 3).....	278
8.8.4 Importing a 2048-bit Host RSA public key.....	280

8.8.5	Importing a 24-byte DES symmetric data encryption key via X9.143 keyblock	282
8.9	Entering passwords using the WFS_CMD_PIN_PASSWORD_ENTRY command.	283
8.9.1	Entering passwords individually to allow secure key parts to be loaded	283
8.9.2	Entering and changing a password	283
9.	Appendix-B (Country Specific WFS_CMD_PIN_ENC_IO protocols)	285
9.1	Luxemburg Protocol	285
9.1.1	WFS_CMD_ENC_IO_LUX_LOAD_APPKEY	287
9.1.2	WFS_CMD_ENC_IO_LUX_GENERATE_MAC	289
9.1.3	WFS_CMD_ENC_IO_LUX_CHECK_MAC	290
9.1.4	WFS_CMD_ENC_IO_LUX_BUILD_PINBLOCK	291
9.1.5	WFS_CMD_ENC_IO_LUX_DECRYPT_TDES	292
9.1.6	WFS_CMD_ENC_IO_LUX_ENCRYPT_TDES	293
9.1.7	Luxemburg-specific Header File	294
9.2	China Protocol	296
9.2.1	WFS_CMD_ENC_IO_CHN_DIGEST	299
9.2.2	WFS_CMD_ENC_IO_CHN_SET_SM2_PARAM	300
9.2.3	WFS_CMD_ENC_IO_CHN_IMPORT_SM2_PUBLIC_KEY	301
9.2.4	WFS_CMD_ENC_IO_CHN_SIGN	303
9.2.5	WFS_CMD_ENC_IO_CHN_VERIFY	305
9.2.6	WFS_CMD_ENC_IO_CHN_EXPORT_SM2_ISSUER_SIGNED_ITEM	306
9.2.7	WFS_CMD_ENC_IO_CHN_GENERATE_SM2_KEY_PAIR	308
9.2.8	WFS_CMD_ENC_IO_CHN_EXPORT_SM2_EPP_SIGNED_ITEM	310
9.2.9	WFS_CMD_ENC_IO_CHN_IMPORT_SM2_SIGNED_SM4_KEY	312
9.2.10	China-specific Header File	315
10.	Appendix-C (Standardized <i>IpszExtra</i> fields)	320
10.1	WFS_INF_PIN_STATUS	320
10.2	WFS_INF_PIN_CAPABILITIES	321
11.	Appendix-D (X9.143 Key Use)	324
12.	Appendix-E (DUKPT)	327
12.1	Default Key Name	327
13.	Appendix-F Diagram Source	328

European Foreword

This CEN Workshop Agreement has been developed in accordance with the CEN-CENELEC Guide 29 “CEN/CENELEC Workshop Agreements – The way to rapid consensus” and with the relevant provisions of CEN/CENELEC Internal Regulations – Part 2. It was approved by a Workshop of representatives of interested parties on 2022-11-08, the constitution of which was supported by CEN following several public calls for participation, the first of which was made on 1998-06-24. However, this CEN Workshop Agreement does not necessarily include all relevant stakeholders.

The final text of this CEN Workshop Agreement was provided to CEN for publication on 2022-11-18.

The following organizations and individuals developed and approved this CEN Workshop Agreement:

- AURIGA SPA
- CIMA SPA
- DIEBOLD NIXDORF SYSTEMS GMBH
- FIS BANKING SOLUTIONS UK LTD (OTS)
- FUJITSU TECHNOLOGY SOLUTIONS
- GLORY LTD
- GRG BANKING EQUIPMENT HK CO LTD
- HITACHI CHANNEL SOLUTIONS CORP
- HYOSUNG TNS INC
- JIANGSU GUO GUANG ELECTRONIC INFORMATION TECHNOLOGY
- KAL
- KEBA HANDOVER AUTOMATION GMBH
- NCR FSG
- NEXUS SOFTWARE
- OBERTHUR CASH PROTECTION
- OKI ELECTRIC INDUSTRY SHENZHEN
- SALZBURGER BANKEN SOFTWARE
- SECURE INNOVATION
- SIGMA SPA

It is possible that some elements of this CEN/CWA may be subject to patent rights. The CEN-CENELEC policy on patent rights is set out in CEN-CENELEC Guide 8 “Guidelines for Implementation of the Common IPR Policy on Patents (and other statutory intellectual property rights based on inventions)”. CEN shall not be held responsible for identifying any or all such patent rights.

The Workshop participants have made every effort to ensure the reliability and accuracy of the technical and non-technical content of CWA 16926-6, but this does not guarantee, either explicitly or implicitly, its correctness. Users of CWA 16926-6 should be aware that neither the Workshop participants, nor CEN can be held liable for damages

or losses of any kind whatsoever which may arise from its application. Users of CWA 16926-6 do so on their own responsibility and at their own risk.

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

Part 19: Biometrics Device Class Interface – Programmer’s Reference

Parts 20 – 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

CWA 16926-65:2023 (E)

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

Part 48: XFS MIB Device Specific Definitions – Biometrics Device Class

Parts 49 – 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Service Provider Interface (SPI) - Programmer's Reference

Part 62: Printer and Scanning Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 75: Card Dispenser Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 76: Barcode Reader Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 77: Item Processing Module Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 78: Biometric Device Class Interface - Migration from Version 3.40 (CWA 16296:2020) to Version 3.50 (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: <https://www.cencenelec.eu/areas-of-work/cen-sectors/digital-society-cen/cwa-download-area/>.

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

Revision History:

3.00	October 18, 2000	Initial Release.
3.10	November 29, 2007	For a description of changes from version 3.00 to version 3.10 see the PIN 3.10 Migration document.
3.20	March 2, 2011	For a description of changes from version 3.10 to version 3.20 see the PIN 3.20 Migration document.
3.30	March 19, 2015	For a description of changes from version 3.20 to version 3.30 see the PIN 3.30 Migration document.
3.40	December 06, 2019	For a description of changes from version 3.30 to version 3.40 see the PIN 3.40 Migration document.
3.50	November 18, 2022	For a description of changes from version 3.40 to version 3.50 see the PIN 3.50 Migration document.

iTeh STANDARD PREVIEW (standards.iteh.ai)

SIST CWA 16926-65:2023

<https://standards.iteh.ai/catalog/standards/sist/21fd83f4-55bd-4c7f-88e2-8fb2ab88dc1a/sist-cwa-16926-65-2023>

1. Introduction

1.1 Background to Release 3.50

The CEN/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 (European Committee for Standardization/Information Society Standardization System) Workshop environment. CEN Workshops aim to arrive at a European consensus on an issue that can be published as a CEN Workshop Agreement (CWA).

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

Release 3.50 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 specification extends the functionality and capabilities of the existing devices covered by the specification:

- Addition of E2E security
- PIN Password Entry

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 for Execute commands or `WFS_ERR_UNSUPP_CATEGORY` error for Info commands is returned to the calling application. An example would be a request from an application to a cash dispenser to retract items where the dispenser hardware does not have that capability; the Service Provider recognizes the command but, since the cash dispenser it is managing is unable to fulfil the request, 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 for Execute commands or `WFS_ERR_INVALID_CATEGORY` error for Info commands 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 error returns to make decisions as to how to use the service.

2. PIN Keypad

This section describes the application program interface for personal identification number keypads (PIN pads) and other encryption/decryption devices. This description includes definitions of the service-specific commands that can be issued, using the **WFSAsyncExecute**, **WFSExecute**, **WFSGetInfo** and **WFSAsyncGetInfo** functions.

This section describes the general interface for the following functions:

- Administration of encryption devices
- Loading of encryption keys
- Encryption / decryption
- Entering Personal Identification Numbers (PINs)
- PIN verification
- PIN block generation (encrypted PIN)
- Clear text data handling
- Function key handling
- PIN presentation to chipcard
- Read and write safety critical Terminal Data from/to HSM
- HSM and Chipcard Authentication
- EMV 4.0 PIN blocks, EMV 4.0 public key loading, static and dynamic data verification

If the PIN pad device has local display capability, display handling should be handled using the Text Terminal Unit (TTU) interface.

The adoption of this specification does not imply the adoption of a specific security standard.

Important Notes:

- This revision of this specification does not define all key management procedures; some key management is still vendor-specific.
- Key space management is customer-specific, and is therefore handled by vendor-specific mechanisms.
- Only numeric PIN pads are handled in this specification.

This specification also supports the Hardware Security Module (HSM), which is necessary for the German ZKA Electronic Purse transactions. Furthermore the HSM stores terminal specific data.

This data will be compared against the message data fields (Sent and Received ISO8583 messages) prior to HSM-MAC generation/verification. HSM-MACs are generated/verified only if the message fields match the data stored.

Keys used for cryptographic HSM functions are stored separate from other keys. This must be considered when importing keys.

This version of PIN pad complies to the current ZKA specification 3.0. It supports loading and unloading against card account for both card types (Type 0 and Type 1) of the ZKA electronic purse. It also covers the necessary functionality for 'Loading against other legal tender'.

Key values are passed to the API as binary hexadecimal values, for example:

0123456789ABCDEF = 0x01 0x23 0x45 0x67 0x89 0xAB 0xCD 0xEF

When hex values are passed to the API within strings, the hex digits 0xA to 0xF can be represented by characters in the ranges 'a' to 'f' or 'A' to 'F'.

The following commands and events were initially added to support the German ZKA standard, but may also be used for other national standards:

- WFS_INF_PIN_HSM_TDATA
- WFS_CMD_PIN_HSM_SET_TDATA
- WFS_CMD_PIN_SECURE_MSG_SEND

CWA 16926-65:2023 (E)

- WFS_CMD_PIN_SECURE_MSG_RECEIVE
- WFS_CMD_PIN_GET_JOURNAL
- WFS_SRVE_PIN_OPT_REQUIRED
- WFS_CMD_PIN_HSM_INIT
- WFS_SRVE_PIN_HSM_TDATA_CHANGED

Certain levels of the PCI EPP security standards specify that if a key encryption key is deleted or replaced, then all keys in the hierarchy under that key encryption key are also removed. Key encryption keys have the WFS_PIN_USEKEYENCKEY type of access. Applications can check impact of key deletion using WFS_INF_PIN_KEY_DETAIL or WFS_INF_PIN_KEY_DETAIL_EX.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

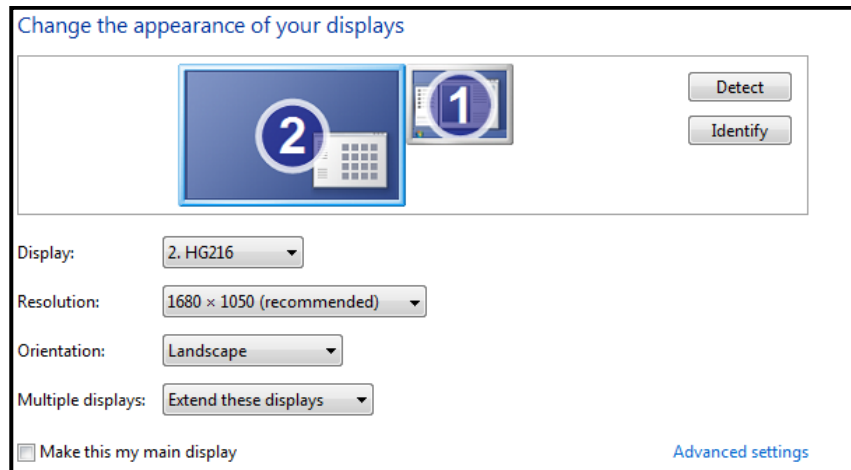
SIST CWA 16926-65:2023

<https://standards.iteh.ai/catalog/standards/sist/21fd83f4-55bd-4c7f-88e2-8fb2ab88dc1a/sist-cwa-16926-65-2023>

2.1 Encrypting Touch Screen (ETS)

An encrypting touch screen device is a touch screen securely attached to a cryptographic device. It can be used as an alternative to an encrypting pin pad (EPP). It supports key management, encryption and decryption.

It is assumed that the ETS is a combined device. It overlays a display monitor which is used to display lead-through for a transaction. It is assumed that the display monitor is part of the Windows desktop, and can be the Windows primary monitor or any other monitor on the desktop. E.g. the following diagram shows 2 monitors extended across the desktop, with monitor 1 being the primary monitor and the ETS being overlaid on monitor 2 whose origin is (-1680.0).



The touch screen can optionally be used as a “mouse” for application purposes, while XFS PIN operations are not in progress or optionally when non-secure XFS PIN commands are in progress.

The CEN interface supports two types of ETS

- Those which activate touch areas defined by the application.
- Those which activate a random variation of touch areas defined by the application.

The Service Provider, when reporting its capabilities, reports the absolute position of the ETS in Windows desktop coordinates. This allows the application to locate the ETS device in a multi-monitor system and relate it to a monitor on the desktop.

At any point in time, a single touch area of the ETS can operate in one of 4 modes:-

- **Mouse mode** - a “touch” simulates a mouse click. This mode is optional. This may not be supported by some ETS devices. Configuration of the click is vendor specific. e.g. WM_LBUTTONDOWN. This is also the mode that, if supported, is active when none of the other modes are active.
- **XFS Data mode** - a “touch” maps to an XFS key and the value of the key is returned in an event (as in clear numeric entry using WFS_CMD_PIN_GET_DATA).
- **XFS PIN mode** - a “touch” maps to an XFS key and the value of the key is returned in an event only if the key pressed is not WFS_PIN_FK_0 through WFS_PIN_FK_9 (as in PIN entry using WFS_CMD_PIN_GET_PIN).
- **XFS Secure mode** - a “touch” maps to an XFS key and the value of the key is returned in an event only if the key pressed is not WFS_PIN_FK_0 through WFS_PIN_FK_9 and not WFS_PIN_FK_A through WFS_PIN_FK_F (as in key entry using WFS_CMD_PIN_SECUREKEY_ENTRY).

The following concepts are introduced to define the relationship between the monitor and the ETS:-

- **Touch Key** – an area of the monitor which reacts to touch in XFS Data, PIN and Secure modes.
- **Touch Frame** – an area of the monitor onto which Touch Keys can be placed. There can be one or more Touch Frames. There may be just one Touch Frame which covers the whole monitor. Areas within a Touch Frame, not defined as a Touch Key, do not react to touch. Generally in XFS PIN and Secure modes, there would be only one Touch Frame covering the whole monitor. An empty Touch Frame disables that part of the monitor.
- **Mouse area** – an area outside of all Touch Frames in which touches behave like a mouse