



# SLOVENSKI STANDARD

## SIST CWA 16926-6:2023

01-marec-2023

---

**Specifikacija vmesnika razširitev za finančne storitve (XFS), izdaja 3.50 - 6. del:  
Vmesnik razreda naprave s tipkovnico PIN - Referenca za programerje**

Extensions for Financial Services (XFS) interface specification Release 3.50 - Part 6:  
PIN Keypad Device Class Interface - Programmer's Reference

iTeh STANDARD PREVIEW  
(standards.iteh.ai)

[SIST CWA 16926-6:2023](https://standards.iteh.ai/catalog/standards/sist/02322595-2023-1100/cwa-16926-6-2023)

Ta slovenski standard je istoveten z: **CWA 16926-6:2022**

---

**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 16926-6:2023**

**en,fr,de**



**CEN****CWA 16926-6****WORKSHOP**

December 2022

**AGREEMENT**

---

**ICS 35.200; 35.240.15; 35.240.40**

English version

## Extensions for Financial Services (XFS) interface specification Release 3.50 - Part 6: PIN Keypad 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-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**

---

© 2022 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.:CWA 16926-6:2022 E

## Table of Contents

---

<b>European Foreword</b> .....	<b>6</b>
<b>1. Introduction</b> .....	<b>10</b>
1.1 Background to Release 3.50 .....	10
1.2 XFS Service-Specific Programming .....	10
<b>2. PIN Keypad</b> .....	<b>12</b>
2.1 Encrypting Touch Screen (ETS).....	14
<b>3. References</b> .....	<b>17</b>
<b>4. Info Commands</b> .....	<b>19</b>
4.1 WFS_INF_PIN_STATUS.....	19
4.2 WFS_INF_PIN_CAPABILITIES .....	23
4.3 WFS_INF_PIN_KEY_DETAIL.....	44
4.4 WFS_INF_PIN_FUNCKEY_DETAIL.....	46
4.5 WFS_INF_PIN_HSM_TDATA .....	49
4.6 WFS_INF_PIN_KEY_DETAIL_EX.....	50
4.7 WFS_INF_PIN_SECUREKEY_DETAIL.....	53
4.8 WFS_INF_PIN_QUERY_LOGICAL_HSM_DETAIL.....	57
4.9 WFS_INF_PIN_QUERY_PCIPTS_DEVICE_ID .....	58
4.10 WFS_INF_PIN_GET_LAYOUT.....	59
4.11 WFS_INF_PIN_KEY_DETAIL_340.....	63
<b>5. Execute Commands</b> .....	<b>65</b>
<b>5.1 Normal PIN Commands</b> .....	<b>65</b>
5.1.1 WFS_CMD_PIN_CRYPT .....	65
5.1.2 WFS_CMD_PIN_IMPORT_KEY .....	68
5.1.3 WFS_CMD_PIN_DERIVE_KEY .....	71
5.1.4 WFS_CMD_PIN_GET_PIN.....	73
5.1.5 WFS_CMD_PIN_LOCAL_DES .....	76
5.1.6 WFS_CMD_PIN_CREATE_OFFSET .....	78
5.1.7 WFS_CMD_PIN_LOCAL_EUROCHEQUE.....	80
5.1.8 WFS_CMD_PIN_LOCAL_VISA.....	82
5.1.9 WFS_CMD_PIN_PRESENT_IDC.....	84
5.1.10 WFS_CMD_PIN_GET_PINBLOCK .....	86
5.1.11 WFS_CMD_PIN_GET_DATA .....	88
5.1.12 WFS_CMD_PIN_INITIALIZATION .....	91
5.1.13 WFS_CMD_PIN_LOCAL_BANKSYS .....	93
5.1.14 WFS_CMD_PIN_BANKSYS_IO .....	94
5.1.15 WFS_CMD_PIN_RESET.....	95
5.1.16 WFS_CMD_PIN_HSM_SET_TDATA.....	96
5.1.17 WFS_CMD_PIN_SECURE_MSG_SEND.....	98
5.1.18 WFS_CMD_PIN_SECURE_MSG_RECEIVE .....	100
5.1.19 WFS_CMD_PIN_GET_JOURNAL .....	102
5.1.20 WFS_CMD_PIN_IMPORT_KEY_EX.....	103
5.1.21 WFS_CMD_PIN_ENC_IO.....	106
5.1.22 WFS_CMD_PIN_HSM_INIT.....	108
5.1.23 WFS_CMD_PIN_SECUREKEY_ENTRY .....	109

5.1.24	WFS_CMD_PIN_GENERATE_KCV .....	112
5.1.25	WFS_CMD_PIN_SET_GUIDANCE_LIGHT .....	113
5.1.26	WFS_CMD_PIN_MAINTAIN_PIN.....	115
5.1.27	WFS_CMD_PIN_KEYPRESS_BEEP .....	116
5.1.28	WFS_CMD_PIN_SET_PINBLOCK_DATA .....	117
5.1.29	WFS_CMD_PIN_SET_LOGICAL_HSM.....	118
5.1.30	WFS_CMD_PIN_IMPORT_KEYBLOCK .....	119
5.1.31	WFS_CMD_PIN_POWER_SAVE_CONTROL .....	120
5.1.32	WFS_CMD_PIN_DEFINE_LAYOUT.....	121
5.1.33	WFS_CMD_PIN_START_AUTHENTICATE.....	122
5.1.34	WFS_CMD_PIN_AUTHENTICATE.....	124
5.1.35	WFS_CMD_PIN_GET_PINBLOCK_EX .....	127
5.1.36	WFS_CMD_PIN_SYNCHRONIZE_COMMAND.....	129
5.1.37	WFS_CMD_PIN_CRYPT_340 .....	130
5.1.38	WFS_CMD_PIN_GET_PINBLOCK_340.....	134
5.1.39	WFS_CMD_PIN_IMPORT_KEY_340.....	136
<b>5.2</b>	<b>Common commands for Remote Key Loading Schemes.....</b>	<b>139</b>
5.2.1	WFS_CMD_PIN_START_KEY_EXCHANGE.....	139
<b>5.3</b>	<b>Remote Key Loading Using Signatures .....</b>	<b>140</b>
5.3.1	WFS_CMD_PIN_IMPORT_RSA_PUBLIC_KEY .....	140
5.3.2	WFS_CMD_PIN_EXPORT_RSA_ISSUER_SIGNED_ITEM.....	143
5.3.3	WFS_CMD_PIN_IMPORT_RSA_SIGNED_DES_KEY .....	145
5.3.4	WFS_CMD_PIN_GENERATE_RSA_KEY_PAIR .....	148
5.3.5	WFS_CMD_PIN_EXPORT_RSA_EPP_SIGNED_ITEM.....	150
<b>5.4</b>	<b>Remote Key Loading with Certificates .....</b>	<b>152</b>
5.4.1	WFS_CMD_PIN_LOAD_CERTIFICATE.....	152
5.4.2	WFS_CMD_PIN_GET_CERTIFICATE.....	153
5.4.3	WFS_CMD_PIN_REPLACE_CERTIFICATE.....	154
5.4.4	WFS_CMD_PIN_IMPORT_RSA_ENCIPHERED_PKCS7_KEY.....	155
5.4.5	WFS_CMD_PIN_LOAD_CERTIFICATE_EX .....	157
5.4.6	WFS_CMD_PIN_IMPORT_RSA_ENCIPHERED_PKCS7_KEY_EX .....	159
<b>5.5</b>	<b>EMV .....</b>	<b>162</b>
5.5.1	WFS_CMD_PIN_EMV_IMPORT_PUBLIC_KEY .....	162
5.5.2	WFS_CMD_PIN_DIGEST .....	165
<b>5.6</b>	<b>Entering and Changing a Password .....</b>	<b>166</b>
5.6.1	WFS_CMD_PIN_PASSWORD_ENTRY .....	166
<b>6.</b>	<b>Events.....</b>	<b>170</b>
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

<b>7. C - Header File .....</b>	<b>183</b>
<b>8. Appendix-A .....</b>	<b>206</b>
<b>8.1 Remote Key Loading Using Signatures .....</b>	<b>207</b>
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 .....	214
8.1.8 Default Keys and Security Item loaded during manufacture.....	215
<b>8.2 Remote Key Loading Using Certificates .....</b>	<b>216</b>
8.2.1 Certificate Exchange and Authentication.....	216
8.2.2 Remote Key Exchange.....	217
8.2.3 Replace Certificate.....	218
8.2.4 Primary and Secondary Certificates.....	219
8.2.5 TR34 BIND To Host.....	220
8.2.6 TR34 Key Transport.....	221
8.2.7 TR34 REBIND To New Host .....	223
8.2.8 TR34 Force REBIND To New Host .....	224
8.2.9 TR34 UNBIND From Host .....	225
8.2.10 TR34 Force UNBIND From Host.....	226
<b>8.3 German ZKA GeldKarte (Deutsche Kreditwirtschaft).....</b>	<b>227</b>
8.3.1 How to use the SECURE_MSG commands.....	227
8.3.2 Protocol WFS_PIN_PROTISOAS .....	228
8.3.3 Protocol WFS_PIN_PROTISOLZ .....	229
8.3.4 Protocol WFS_PIN_PROTISOPS.....	230
8.3.5 Protocol WFS_PIN_PROTCHIPZKA .....	231
8.3.6 Protocol WFS_PIN_PROTRAWDATA.....	232
8.3.7 Protocol WFS_PIN_PROTPBM .....	233
8.3.8 Protocol WFS_PIN_PROTHSMLDI .....	234
8.3.9 Protocol WFS_PIN_PROTGENAS .....	235
8.3.10 Protocol WFS_PIN_PROTCHIPINCHG.....	239
8.3.11 Protocol WFS_PIN_PROTPINCOMP.....	240
8.3.12 Protocol WFS_PIN_PROTISOPINCHG .....	242
8.3.13 Command Sequence.....	243
<b>8.4 EMV Support.....</b>	<b>250</b>
8.4.1 Keys loading.....	250
8.4.2 PIN Block Management .....	252
8.4.3 SHA-1 Digest.....	253
<b>8.5 French Cartes Bancaires.....</b>	<b>254</b>
8.5.1 Data Structure for WFS_CMD_PIN_ENC_IO .....	254
8.5.2 Command Sequence.....	256
<b>8.6 Secure Key Entry .....</b>	<b>258</b>
8.6.1 Keyboard Layout.....	258
8.6.2 Command Usage - WFS_CMD_PIN_SECUREKEY_DETAIL and WFS_CMD_PIN_IMPORT_KEY.....	262
8.6.3 Command Usage - WFS_INF_PIN_GET_LAYOUT and WFS_CMD_PIN_IMPORT_KEY_340 .....	263
<b>8.7 WFS_PIN_USERRESTRICTEDKEYENCKEY key usage.....</b>	<b>264</b>
8.7.1 Command Usage .....	264
<b>8.8 WFS_CMD_PIN_IMPORT_KEY_340 command Input/Output Parameters.....</b>	<b>267</b>
8.8.1 Importing a 3DES 16-byte terminal master key using signature-based remote key loading (SRKL): 268	
8.8.2 Importing a 16-byte DES key for PIN encryption with a key check value in the input .....	270
8.8.3 Importing a 16-byte DES key for MACing (MAC Algorithm 3).....	272
8.8.4 Importing a 2048-bit Host RSA public key.....	274

8.8.5	Importing a 24-byte DES symmetric data encryption key via X9.143 keyblock .....	276
<b>8.9</b>	<b>Entering passwords using the WFS_CMD_PIN_PASSWORD_ENTRY command. ....</b>	<b>277</b>
8.9.1	Entering passwords individually to allow secure key parts to be loaded .....	277
8.9.2	Entering and changing a password .....	277
<b>9.</b>	<b>Appendix-B (Country Specific WFS_CMD_PIN_ENC_IO protocols) .....</b>	<b>279</b>
<b>9.1</b>	<b>Luxemburg Protocol .....</b>	<b>279</b>
9.1.1	WFS_CMD_ENC_IO_LUX_LOAD_APPKEY .....	281
9.1.2	WFS_CMD_ENC_IO_LUX_GENERATE_MAC .....	283
9.1.3	WFS_CMD_ENC_IO_LUX_CHECK_MAC .....	284
9.1.4	WFS_CMD_ENC_IO_LUX_BUILD_PINBLOCK .....	285
9.1.5	WFS_CMD_ENC_IO_LUX_DECRYPT_TDES .....	286
9.1.6	WFS_CMD_ENC_IO_LUX_ENCRYPT_TDES .....	287
9.1.7	Luxemburg-specific Header File .....	288
<b>9.2</b>	<b>China Protocol .....</b>	<b>290</b>
9.2.1	WFS_CMD_ENC_IO_CHN_DIGEST .....	293
9.2.2	WFS_CMD_ENC_IO_CHN_SET_SM2_PARAM .....	294
9.2.3	WFS_CMD_ENC_IO_CHN_IMPORT_SM2_PUBLIC_KEY .....	295
9.2.4	WFS_CMD_ENC_IO_CHN_SIGN .....	297
9.2.5	WFS_CMD_ENC_IO_CHN_VERIFY .....	299
9.2.6	WFS_CMD_ENC_IO_CHN_EXPORT_SM2_ISSUER_SIGNED_ITEM .....	300
9.2.7	WFS_CMD_ENC_IO_CHN_GENERATE_SM2_KEY_PAIR .....	302
9.2.8	WFS_CMD_ENC_IO_CHN_EXPORT_SM2_EPP_SIGNED_ITEM .....	303
9.2.9	WFS_CMD_ENC_IO_CHN_IMPORT_SM2_SIGNED_SM4_KEY .....	305
9.2.10	China-specific Header File .....	308
<b>10.</b>	<b>Appendix-C (Standardized <i>IpszExtra</i> fields) .....</b>	<b>313</b>
10.1	WFS_INF_PIN_STATUS .....	313
10.2	WFS_INF_PIN_CAPABILITIES .....	314
<b>11.</b>	<b>Appendix-D (X9.143 Key Use) .....</b>	<b>317</b>
<b>12.</b>	<b>Appendix-E (DUKPT) .....</b>	<b>320</b>
12.1	Default Key Name .....	320
<b>13.</b>	<b>Appendix-F Diagram Source .....</b>	<b>321</b>

**CWA 16926-6:2022 (E)****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 16926-6:2022 (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 16926-6:2023](https://standards.iteh.ai/catalog/standards/sist/025b3994-bfd7-44c9-8cdf-5bf8e29de166/sist-cwa-16926-6-2023)

<https://standards.iteh.ai/catalog/standards/sist/025b3994-bfd7-44c9-8cdf-5bf8e29de166/sist-cwa-16926-6-2023>

## 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.

## iTeh STANDARD PREVIEW (standards.iteh.ai)

[SIST CWA 16926-6:2023](https://standards.iteh.ai/catalog/standards/sist/025b3994-bfd7-44c9-8cdf-5bf8e29de166/sist-cwa-16926-6-2023)

<https://standards.iteh.ai/catalog/standards/sist/025b3994-bfd7-44c9-8cdf-5bf8e29de166/sist-cwa-16926-6-2023>

## 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

- 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 16926-6:2023](https://standards.iteh.ai/catalog/standards/sist/025b3994-bfd7-44c9-8cdf-5bf8e29de166/sist-cwa-16926-6-2023)

<https://standards.iteh.ai/catalog/standards/sist/025b3994-bfd7-44c9-8cdf-5bf8e29de166/sist-cwa-16926-6-2023>