



SLOVENSKI STANDARD

SIST CWA 16926-9:2023

01-marec-2023

**Specifikacija vmesnika razširitev za finančne storitve (XFS), izdaja 3.50 - 9. del:
Vmesnik razreda naprave tekstovne terminalne enote - Referenca za programerje**

Extensions for Financial Services (XFS) interface specification Release 3.50 - Part 9:
Text Terminal Unit Device Class Interface - Programmer's Reference

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[SIST CWA 16926-9:2023](https://standards.iteh.ai/standards/sist/7261-6520-45-2-1023-6204-64fb3b45a4d5/sist-cwa-16926-9-2023)

Ta slovenski standard je istoveten z: **CWA 16926-9:2022**

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 16926-9:2023

en,fr,de

CEN**CWA 16926-9****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 9: Text Terminal Unit 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 16926-9:2022 E

Table of Contents

European Foreword.....	4
1. Introduction.....	8
1.1 Background to Release 3.50	8
1.2 XFS Service-Specific Programming	8
2. Text Terminal Units	10
3. References	11
4. Info Commands	12
4.1 WFS_INF_TTU_STATUS.....	12
4.2 WFS_INF_TTU_CAPABILITIES.....	15
4.3 WFS_INF_TTU_FORM_LIST.....	18
4.4 WFS_INF_TTU_QUERY_FORM.....	19
4.5 WFS_INF_TTU_QUERY_FIELD	20
4.6 WFS_INF_TTU_KEY_DETAIL.....	22
5. Execute Commands	24
5.1 WFS_CMD_TTU_BEEP	24
5.2 WFS_CMD_TTU_CLEARSCREEN.....	25
5.3 WFS_CMD_TTU_DISPLIGHT.....	26
5.4 WFS_CMD_TTU_SET_LED.....	27
5.5 WFS_CMD_TTU_SET_RESOLUTION.....	28
5.6 WFS_CMD_TTU_WRITE_FORM	29
5.7 WFS_CMD_TTU_READ_FORM	30
5.8 WFS_CMD_TTU_WRITE	32
5.9 WFS_CMD_TTU_READ.....	34
5.10 WFS_CMD_TTU_RESET.....	37
5.11 WFS_CMD_TTU_DEFINE_KEYS.....	38
5.12 WFS_CMD_TTU_POWER_SAVE_CONTROL.....	40
5.13 WFS_CMD_TTU_SET_LED_EX.....	41
5.14 WFS_CMD_TTU_SYNCHRONIZE_COMMAND	42
6. Events.....	43
6.1 WFS_EXEE_TTU_FIELDERRORE.....	43
6.2 WFS_EXEE_TTU_FIELDWARNING.....	44
6.3 WFS_EXEE_TTU_KEY	45
6.4 WFS_SRVE_TTU_DEVICEPOSITION.....	46
6.5 WFS_SRVE_TTU_POWER_SAVE_CHANGE.....	47
7. Form and Field Definitions	48

7.1	Definition Syntax	48
7.2	XFS form/media definition files in multi-vendor environments	49
7.3	Form Definition	50
7.4	Field Definition	51
8.	C - Header file	53

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST CWA 16926-9:2023

<https://standards.iteh.ai/catalog/standards/sist/72016504-d9e2-4068-8204-64fb3b45a4d5/sist-cwa-16926-9-2023>

CWA 16926-9:2022 (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
- KEB A 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 16926-9, but this does not guarantee, either explicitly or implicitly, its correctness. Users of CWA 16926-9 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 16926-9 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 16926-9:2022 (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 16926:2020) 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 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 75: Card Dispenser Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 76: Barcode Reader Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) to Version 3.50 (this CWA) - Programmer's Reference

Part 77: Item Processing Module Device Class Interface - Migration from Version 3.40 (CWA 16926:2020) 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.cen.eu/work/Sectors/Digital_society/Pages/WSXFS.aspx.

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:

b3.10	November 29, 2007	Initial Release.
3.20	March 2, 2011	For a description of changes from version 3.10 to version 3.20 see the TTU 3.20 Migration document.
3.30	March 19, 2015	For a description of changes from version 3.20 to version 3.30 see the TTU 3.30 Migration document.
3.40	December 06, 2019	For a description of changes from version 3.30 to version 3.40 see the TTU 3.40 Migration document.
3.50	November 18, 2022	For a description of changes from version 3.40 to version 3.50 see the TTU 3.50 Migration document.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST CWA 16926-9:2023

<https://standards.iteh.ai/catalog/standards/sist/72016504-d9e2-4068-8204-64fb3b45a4d5/sist-cwa-16926-9-2023>

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. Notable enhancements include:

- 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 the 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 16926-9:2023

<https://standards.iteh.ai/catalog/standards/sist/72016504-d9e2-4068-8204-64fb3b45a4d5/sist-cwa-16926-9-2023>

2. Text Terminal Units

This specification describes the functionality of the services provided by text terminal unit (TTU) services under XFS, by defining the service-specific commands that can be issued, using the **WFSGetInfo**, **WFSAsyncGetInfo**, **WFSExecute** and **WFSAsyncExecute** functions.

This section describes the functions provided by a generic Text Terminal Unit (TTU) service. A Text Terminal Unit is a text i/o device, which applies both to ATM operator panels and to displays incorporated in devices such as PIN pads and printers. This service allows for the following categories of functions:

- Forms oriented input and output
- Direct display output
- Keyboard input
- LED settings and control

All position indexes are zero based, where column zero, row zero is the top-leftmost position.

If the device has no shift key, the **WFS_CMD_TTU_READ_FORM** and **WFS_CMD_TTU_READ** commands will return only upper case letters. If the device has a shift key, these commands return upper and lower case letters as governed by the user's use of the shift key.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST CWA 16926-9:2023

<https://standards.iteh.ai/catalog/standards/sist/72016504-d9e2-4068-8204-64fb3b45a4d5/sist-cwa-16926-9-2023>

3. References

- | |
|--|
| 1. XFS Application Programming Interface (API)/Service Provider Interface (SPI), Programmer's Reference
Revision 3.50 |
|--|

iTeh STANDARD PREVIEW
(standards.iteh.ai)

SIST CWA 16926-9:2023

<https://standards.iteh.ai/catalog/standards/sist/72016504-d9e2-4068-8204-64fb3b45a4d5/sist-cwa-16926-9-2023>

4. Info Commands

4.1 WFS_INF_TTU_STATUS

Description This command reports the full range of information available, including the information that is provided by the Service Provider.

Input Param None.

Output Param LPWFSTTUSTATUS lpStatus;

```
typedef struct _wfs_ttu_status
{
    WORD fwDevice;
    WORD wKeyboard;
    WORD wKeylock;
    WORD wLEDs[WFS_TTU_LEDS_MAX];
    WORD wDisplaySizeX;
    WORD wDisplaySizeY;
    LPSTR lpszExtra;
    WORD wDevicePosition;
    USHORT usPowerSaveRecoveryTime;
    LPWFSTTULEDEX lpLEDEX;
    WORD wAntiFraudModule;
} WFSTTUSTATUS, *LPWFSTTUSTATUS;
```

fwDevice

Specifies the state of the text terminal unit as one of the following flags:

Value	Meaning
WFS_TTU_DEVONLINE	The device is online (i.e. powered on and operable).
WFS_TTU_DEVOFFLINE	The device is offline (e.g. the operator has taken the device offline by turning a switch).
WFS_TTU_DEVPOWEROFF	The device is powered off or physically not connected.
WFS_TTU_DEVBUSY	The device is busy and unable to process an execute command at this time.
WFS_TTU_DEVNODEVICE	There is no device intended to be there; e.g. this type of self service machine does not contain such a device or it is internally not configured.
WFS_TTU_DEVHWERROR	The device is inoperable due to a hardware error.
WFS_TTU_DEVUSERERROR	The device is inoperable because a person is preventing proper device operation.
WFS_TTU_DEVFRAUDATTEMPT	The device is present but is inoperable because it has detected a fraud attempt.
WFS_TTU_DEVPOTENTIALFRAUD	The device has detected a potential fraud attempt and is capable of remaining in service. In this case the application should make the decision as to whether to take the device offline.

wKeyboard

Specifies the state of the keyboard in the text terminal unit as one of the following flags:

Value	Meaning
WFS_TTU_KBDON	The keyboard is activated.
WFS_TTU_KBDOFF	The keyboard is not activated.
WFS_TTU_KBDNA	The keyboard is not available.

wKeylock

Specifies the state of the keyboard lock of the text terminal unit as one of the following flags:

Value	Meaning
WFS_TTU_KBDLOCKON	The keyboard lock switch is activated.
WFS_TTU_KBDLOCKOFF	The keyboard lock switch is not activated.
WFS_TTU_KBDLOCKNA	The keyboard lock switch is not available.

wLEDs [WFS_TTU_LEDS_MAX]

Specifies the state of the LEDs. The maximum LED index is WFS_TTU_LEDS_MAX -1. The number of available LEDs can be retrieved with the WFS_INF_TTU_CAPABILITIES info command. This field is only provided for backwards compatibility; the *lpLEDEx* parameter should instead be used to retrieve the LED status. All member elements in this array are specified as one of the following flags:

Value	Meaning
WFS_TTU_LEDNA	The status is not available.
WFS_TTU_LEDOFF	The LED is turned off.
WFS_TTU_LEDSLOWFLASH	The LED is blinking slowly.
WFS_TTU_LEDMEDIUMFLASH	The LED is blinking medium frequency.
WFS_TTU_LEDQUICKFLASH	The LED is blinking quickly.
WFS_TTU_LEDCONTINUOUS	The light is turned on continuous (steady).

wDisplaySizeX

Specifies the horizontal size of the display of the text terminal unit (the number of columns that can be displayed).

wDisplaySizeY

Specifies the vertical size of the display of the text terminal unit (the number of rows that can be displayed).

lpszExtra

Pointer to a list of vendor-specific, or any other extended, information. The information is returned as a series of “key=value” strings so that it is easily extensible by Service Providers. Each string is null-terminated, with the final string terminating with two null characters. An empty list may be indicated by either a NULL pointer or a pointer to two consecutive null characters.

wDevicePosition

Specifies the device position. The device position value is independent of the *fwDevice* value, e.g. when the device position is reported as WFS_TTU_DEVICENOTINPOSITION, *fwDevice* can have any of the values defined above (including WFS_TTU_DEVONLINE or WFS_TTU_DEVOFFLINE). This value is one of the following values:

Value	Meaning
WFS_TTU_DEVICEINPOSITION	The device is in its normal operating position, or is fixed in place and cannot be moved.
WFS_TTU_DEVICENOTINPOSITION	The device has been removed from its normal operating position.
WFS_TTU_DEVICEPOSUNKNOWN	Due to a hardware error or other condition, the position of the device cannot be determined.
WFS_TTU_DEVICEPOSNOTSUPP	The physical device does not have the capability of detecting the position.

usPowerSaveRecoveryTime

Specifies the actual number of seconds required by the device to resume its normal operational state from the current power saving mode. This value is zero if either the power saving mode has not been activated or no power save control is supported.

lpLEDEx

Pointer to a WFSTTULEDEX structure that specifies the states of the LEDs. If there is no LED available this will be a NULL pointer.

```
typedef struct _wfs_ttu_led_ex
{
    USHORT                usNumOfLEDs;
    LPDWORD               lpdwLEDs;
} WFSTTULEDEX, *LPWFSTTULEDEX;
```