

**CEN**

**CWA 13449-10**

**WORKSHOP**

**AGREEMENT**

December 1998

---

ICS 35.200;35.240.40

English version

**Extensions for Financial Services (XFS) interface specification -  
Part 10: Sensors and Indicators Unit Device Class Interface -  
Programmer's Interface**

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 Central Secretariat 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, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.



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

**Central Secretariat: rue de Stassart, 36 B-1050 Brussels**

## Contents

---

<b>FOREWORD</b> .....	<b>3</b>
<b>0. INTRODUCTION</b> .....	<b>4</b>
<b>1. XFS SERVICE-SPECIFIC PROGRAMMING</b> .....	<b>5</b>
<b>2. SENSORS AND INDICATORS UNIT</b> .....	<b>6</b>
<b>3. INFO COMMANDS</b> .....	<b>7</b>
3.1 WFS_INF_SIU_STATUS.....	7
3.2 WFS_INF_SIU_CAPABILITIES.....	12
<b>4. EXECUTE COMMANDS</b> .....	<b>17</b>
4.1 WFS_CMD_SIU_ENABLE_EVENTS .....	17
4.2 WFS_CMD_SIU_SET_PORTS.....	22
4.3 WFS_CMD_SIU_SET_DOOR.....	25
4.4 WFS_CMD_SIU_SET_INDICATOR.....	26
4.5 WFS_CMD_SIU_SET_AUXILIARY .....	27
4.6 WFS_CMD_SIU_SET_GUIDLIGHT.....	28
<b>5. EVENTS</b> .....	<b>29</b>
5.1 WFS_SRVE_SIU_PORT_STATUS.....	29
5.2 WFS_EXEE_SIU_PORT_ERROR.....	30
<b>6. C - HEADER FILE</b> .....	<b>33</b>

## Foreword

---

This CWA is revision 2.0 of the XFS interface specification. Release 2.0 extends the scope of the XFS interface specification to include both the self service/ATM environment as well as the branch environment. The new specification now fully supports cameras, deposit units, identification cards, PIN pads, sensors and indicator units, text terminals, cash dispenser modules and a wide variety of printing mechanisms.

This specification was originally developed by the Banking Solutions Vendor Council (BSVC), and is endorsed by the CEN/ISSS Workshop on XFS. This Workshop gathers both suppliers (among others the BSVC members) 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.

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 2.00.

This CWA is supplemented by a set of release notes, which are available from the CEN/ISSS Secretariat (an on-line version of these release notes is available from <http://www.cenorm.be/iss/Workshop/XFS/release-notes.htm>).

## 0. Introduction

This is part 10 of the multi-part CWA 13449, describing Release 2.0 of the XFS interface specification.

The full CWA 13449 "Extensions for Financial Services (XFS) interface specification" consists of the following parts:

**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**

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 from the CEN/ISSS Secretariat (contact [iss@cenorm.be](mailto:iss@cenorm.be) or download from <http://www.cenorm.be/iss/Workshop/XFS/release-notes.htm>).

The information in this document originally contributed by members of the Banking Solutions Vendor Council and endorsed by the CEN/ISSS Workshop on XFS, 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.

The XFS specifications are now further developed in the CEN/ISSS Workshop on XFS. CEN/ISSS Workshops are open to all interested parties offering to contribute. Parties interested in participating should contact the CEN/ISSS Secretariat ([iss@cenorm.be](mailto:iss@cenorm.be)).

A Software Development Kit (SDK) which supplies the components and tools to allow the implementation of compliant applications and services is available from Microsoft<sup>1</sup>.

To the extent that date processing occurs, all XFS Workshop participants agree that the XFS specifications are Year 2000 compliant.

### Revision History:

1.0	May 24, 1993	Initial release of API and SPI specification
1.11	February 3, 1995	Separation of specification into separate documents for API/SPI and service class definitions, with updates
2.00	November 11, 1996 October 6, 1998	Updated release encompassing self-service environment. WOSA/XFS Release 2.00 as originally developed by the BSVC, has been formally accepted as a CEN Workshop Agreement by the CEN/ISSS XFS Workshop and the name WOSA/XFS has been changed into XFS. In spite of the name change, certain occurrences of WOSA/XFS however still appear in the documentation, for compatibility reasons

<sup>1</sup> Microsoft is a registered trademark, and Windows and Windows NT are trademarks of Microsoft Corporation

## 1. 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 Extensions for Financial Services specification is to standardize command 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 the union 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\_UNSUPP\_COMMAND** error is returned to the calling application. An example would be a request from an application to a cash dispenser to dispense coins; the service provider recognizes the command but, since the cash dispenser it is managing dispenses only notes, returns this error.
- The requested capability is *not* defined for the class of service providers by the XFS specification. In this case, a **WFS\_ERR\_INVALID\_COMMAND** error is returned to the calling application.

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

## 2. Sensors and Indicators Unit

---

This specification describes the functionality of the services provided by the Sensors and Indicators Unit (SIU) 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 Sensors and Indicators Unit service. This service allows for the operation of the following categories of ports:

- Door sensors, such as cabinet, safe or vandal shield doors;
- Alarm sensors, such as tamper, seismic or heat sensors;
- Generic sensors, such as proximity or ambient light sensors;
- Key switch sensors, such as the ATM operator switch;
- Lamp/sign indicators, such as fascia light or audio indicators;
- Auxiliary indicators.

In self-service devices, the sensors and indicators unit is capable of dealing with external sensors, such as door switches, locks, alarms and proximity sensors, as well as external indicators, such as turning on lamps or heating.

## 3. Info Commands

### 3.1 WFS\_INF\_SIU\_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** LPWFSSIUSTATUS lpStatus;

```
typedef struct _wfs_siu_status
{
    WORD          fwDevice;
    WORD          fwSensors [WFS_SIU_SENSORS_SIZE];
    WORD          fwDoors [WFS_SIU_DOORS_SIZE];
    WORD          fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD          fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD          fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR         lpzExtra;
} WFS_SIU_STATUS, * LPWFSSIUSTATUS;
```

*fwDevice*

Specifies the state of the Sensors and Indicators Unit device as one of the following flags:

Value	Meaning
WFS_SIU_DEVONLINE	The device is on-line. The device is present and operational (i.e. not busy processing a request and does not have a hardware error).
WFS_SIU_DEVOFFLINE	The device is off-line. The device is present and powered on but it is not operational (e.g. a switch may have been used to change it to an off-line state).
WFS_SIU_DEVPOWEROFF	The device is powered off. The device is present, but is currently powered off.
WFS_SIU_DEVBUSY	The device is busy processing a request. The device is present and an EXECUTE request is currently being processing.
WFS_SIU_DEVNODEVICE	There is no device connected.
WFS_SIU_DEVHWERROR	The device is inoperable due to a hardware error. The device is present but a hardware fault prevents it from being used.
WFS_SIU_DEVUSERERROR	The device is present but a person is preventing proper operation. The application should suspend the device operation or remove the device from service until the service provider generates a device state change event indicating the condition of the device has changed i.e. the error is removed (WFS_SIU_DEVONLINE) or a permanent error condition has occurred (WFS_SIU_DEVHWERROR).

*fwSensors [...]*

Specifies the state of the Sensors. A number of sensor types are defined below. Vendor specific sensors are defined starting from the end of the array. The maximum sensor index is WFS\_SIU\_SENSORS\_MAX.

*fwSensors [WFS\_SIU\_OPERATORSWITCH]*

Specifies the state of the Operator Switch(es). This switch is used to tell the terminal if an Operator/Supervisor wants to change the state from Run to Operators/Supervisors mode or vice versa. The **Run** mode is used for normal consumer operations/transactions. The **Maintenance** mode is used when replenish the terminal. The **Supervisor** mode is used

when operating the terminal for service and testing. Supervisor mode has higher priority than maintenance mode. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_RUN	The switch is in <b>Run</b> mode.
WFS_SIU_MAINTENANCE	The switch is in <b>Maintenance</b> mode.
WFS_SIU_SUPERVISOR	The switch is in <b>Supervisor</b> mode.

*fwSensors [WFS\_SIU\_TAMPER]*

Specifies the state of the Tamper Sensor for the terminal. This sensor indicates whether the terminal has been tampered with (such as a burglar attempt). Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	There is <b>no</b> indication of a tampering attempt.
WFS_SIU_ON	There <b>has been</b> a tampering attempt.

*fwSensors [WFS\_SIU\_INTTAMPER]*

Specifies the state of the Tamper Sensor for the internal alarm. This sensor indicates whether the internal alarm has been tampered with (such as a burglar attempt). Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	There is <b>no</b> indication of a tampering attempt.
WFS_SIU_ON	There <b>has been</b> a tampering attempt.

*fwSensors [WFS\_SIU\_SEISMIC]*

Specifies the state of the Seismic Sensor. This sensor indicates whether the terminal has been shaken (e.g. burglar attempt or seismic activity). Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	The seismic activity has <b>not</b> been high enough to trigger the sensor.
WFS_SIU_ON	The seismic or other activity <b>has</b> triggered the sensor.

*fwSensors [WFS\_SIU\_HEAT]*

Specifies the state of the Heat Sensor. This sensor is triggered by excessive heat (fire) near the terminal. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	The heat has <b>not</b> been high enough to trigger the sensor.
WFS_SIU_ON	The heat <b>has been</b> high enough to trigger the sensor.

*fwSensors [WFS\_SIU\_PROXIMITY]*

Specifies the state of the Proximity Sensor. This sensor is triggered by movements around the terminal. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_NOT_PRESENT	The sensor can <b>not</b> sense any people around the terminal.
WFS_SIU_PRESENT	The sensor is showing that there is <b>someone</b> present at the terminal.



*fwSensors [WFS\_SIU\_AMBLIGHT]*

Specifies the state of the Ambient Light Sensor. This sensor indicates the level of ambient light around the terminal. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_VERY_LIGHT	The level of light is: <b>very light</b>
WFS_SIU_LIGHT	The level of light is: <b>light</b>
WFS_SIU_MEDIUM_LIGHT	The level of light is: <b>medium light</b>
WFS_SIU_DARK	The level of light is: <b>dark</b>
WFS_SIU_VERY_DARK	The level of light is: <b>very dark</b>

*fwDoors [...]*

Specifies the state of the Doors. A number of door types are defined below. Vendor specific doors are defined starting from the end of the array. The maximum door index is WFS\_SIU\_DOORS\_MAX.

*fwDoors [WFS\_SIU\_CABINET]*

Specifies the state of the Cabinet Doors. Cabinet Doors are doors that open up for consumables, and hardware that does not have to be in a secure place. Specified as one of the following flags.

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OPEN	At least one of the Cabinet Doors is <b>open</b> .
WFS_SIU_CLOSED	The Cabinet Doors are <b>closed</b> .
WFS_SIU_LOCKED	The Cabinet Doors are closed and <b>locked</b> .
WFS_SIU_BOLTED	The Cabinet Doors are closed, locked and <b>bolted</b> .

*fwDoors [WFS\_SIU\_SAFE]*

Specifies the state of the Safe Doors. Safe Doors are doors that open up for secure hardware, such as the note dispenser, the security device, etc. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OPEN	At least one of the Safe Doors is <b>open</b> .
WFS_SIU_CLOSED	The Safe Doors are <b>closed</b> .
WFS_SIU_LOCKED	The Safe Doors are closed and <b>locked</b> .
WFS_SIU_BOLTED	The Safe Doors are closed, locked and <b>bolted</b> .

*fwDoors [WFS\_SIU\_VANDALSHIELD]*

Specifies the state of the Vandal Shield. The Vandal Shield is a door that open up for consumer access to the terminal. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OPEN	The Vandal Shield is <b>open</b> .
WFS_SIU_CLOSED	The Vandal Shield is <b>closed</b> .
WFS_SIU_LOCKED	The Vandal Shield is closed and <b>locked</b> .
WFS_SIU_SERVICE	The Vandal Shield is in <b>service</b> position.
WFS_SIU_KEYBOARD	The Vandal Shield position permits access to the keyboard.
WFS_SIU_AJAR	The Vandal Shield is ajar.
WFS_SIU_JAMMED	The Vandal Shield is jammed.

*fwIndicators [...]*

Specifies the state of the Status Indicators. A number of Status Indicator types are defined below. Vendor specific indicators are defined starting from the end of the array. The maximum indicator index is WFS\_SIU\_INDICATORS\_MAX.

*fwIndicators [WFS\_SIU\_OPENCLOSE]*

Specifies the state of the Open/Closed Indicator as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_CLOSED	The terminal is <b>closed</b> for a consumer.
WFS_SIU_OPEN	The terminal is <b>open</b> to be used by a consumer.

*fwIndicators [WFS\_SIU\_FASCIALIGHT]*

Specifies the state of the Fascia Light as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	The Fascia Light is turned <b>off</b> .
WFS_SIU_ON	The Fascia Light is turned <b>on</b> .

*fwIndicators [WFS\_SIU\_AUDIO]*

Specifies the state of the Audio Indicator as one of the following flags of type A and B, or as WFS\_SIU\_CONTINUOUS in combination with one of the flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	The status is not available.	A
WFS_SIU_OFF	The Audio Indicator is turned <b>off</b> .	A
WFS_SIU_KEYPRESS	The Audio Indicator sounds a <b>key click</b> signal.	B
WFS_SIU_EXCLAMATION	The Audio Indicator sounds a <b>exclamation</b> signal.	B
WFS_SIU_WARNING	The Audio Indicator sounds a <b>warning</b> signal.	B
WFS_SIU_ERROR	The Audio Indicator sounds a <b>error</b> signal.	B
WFS_SIU_CRITICAL	The Audio Indicator sounds a <b>critical</b> signal.	B
WFS_SIU_CONTINUOUS	The Audio Indicator sound is turned on continuously.	C

*fwIndicators [WFS\_SIU\_HEATING]*

Specifies the state of the internal heating as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	The Heating is turned <b>off</b> .
WFS_SIU_ON	The Heating is turned <b>on</b> .

*fwAuxiliaries [...]*

Specifies the state of the Auxiliary indicators. A number of Auxiliary indicator types are defined below. Vendor specific Auxiliaries are defined starting from the end of the array. The maximum auxiliary index is WFS\_SIU\_AUXILIARIES\_MAX.

*fwAuxiliaries [WFS\_SIU\_VOLUME]*

Specifies the value of the volume control. The value of volume control is defined in an interval from 1 to 1000 where 1 is the lowest volume level and 1000 is the highest volume level. The interval is defined in logarithmic steps, e.g. a volume control on a radio. Note: The volume control field is handled as unsigned short.

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
1, ..., 1000	The volume level. This field is handled as an unsigned short.

*fwAuxiliaries [WFS\_SIU\_UPS]*

Specifies the state of the Uninterruptable Power Supply device as WFS\_SIU\_NOT\_AVAILABLE or as a combination of the following flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	There is no UPS available.	A
WFS_SIU_AVAILABLE	The UPS is available	B

WFS_SIU_LOW	The charge level of the UPS is low	B
WFS_SIU_ENGAGED	The UPS is engaged	B
WFS_SIU_POWERING	The UPS is powering the system. The main power supply is off.	B
WFS_SIU_RECOVERED	The UPS was engaged when the main power went off	B

*fwGuidLights [...]*

Specifies the state of the Guidance Light Indicators. A number of guidance light types are defined below. Vendor specific guidance lights are defined starting from the end of the array. The maximum guidance light index is WFS\_SIU\_GUIDLIGHTS\_MAX. All member elements in this array are specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	The status is not available.
WFS_SIU_OFF	The light is turned <b>off</b> .
WFS_SIU_SLOW_FLASH	The light is <b>blinking slowly</b> .
WFS_SIU_MEDIUM_FLASH	The light is <b>blinking medium frequency</b> .
WFS_SIU_QUICK_FLASH	The light is <b>blinking quickly</b> .
WFS_SIU_CONTINUOUS	The light is turned on <b>continuous</b> (steady).

*fwGuidLights [WFS\_SIU\_CARDUNIT]*

Specifies the state of the Guidance Light Indicator on the Card Unit (IDC).

*fwGuidLights [WFS\_SIU\_PINPAD]*

Specifies the state of the Guidance Light Indicator on the PIN pad unit.

*fwGuidLights [WFS\_SIU\_NOTESDISPENSER]*

Specifies the state of the Guidance Light Indicator on the note dispenser unit.

*fwGuidLights [WFS\_SIU\_COINDISPENSER]*

Specifies the state of the Guidance Light Indicator on the coin dispenser unit.

*fwGuidLights [WFS\_SIU\_RECEIPTPRINTER]*

Specifies the state of the Guidance Light Indicator on the receipt printer unit.

*fwGuidLights [WFS\_SIU\_PASSBOOKPRINTER]*

Specifies the state of the Guidance Light Indicator on the passbook printer unit.

*fwGuidLights [WFS\_SIU\_ENVDEPOSITORY]*

Specifies the state of the Guidance Light Indicator on the envelope depository unit.

*fwGuidLights [WFS\_SIU\_CHEQUEUNIT]*

Specifies the state of the Guidance Light Indicator on the cheque processing unit.

*fwGuidLights [WFS\_SIU\_BILLACCEPTOR]*

Specifies the state of the Guidance Light Indicator on the bill acceptor unit.

*fwGuidLights [WFS\_SIU\_ENVDISPENSER]*

Specifies the state of the Guidance Light Indicator on the envelope dispenser unit.

*lpszExtra*

Specifies 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 will be null-terminated, with the final string terminating with two null characters.

**Error Codes** There are no additional error codes generated by this command.

**Comments** Applications which require or expect specific information to be present in the *lpzExtra* parameter may not be device or vendor-independent.

### 3.2 WFS\_INF\_SIU\_CAPABILITIES

---

**Description** This command is used to retrieve the capabilities of the Sensors and Indicators Unit.

**Input Param** None.

**Output Param** LPWFSSIUCAPS lpCaps;

```
typedef struct _wfs_siu_caps
{
    WORD          wClass;
    WORD          fwType;
    WORD          fwSensors [WFS_SIU_SENSORS_SIZE];
    WORD          fwDoors [WFS_SIU_DOORS_SIZE];
    WORD          fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD          fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD          fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR         lpzExtra;
} WFS_SIU_CAPS, * LPWFSSIUCAPS;
```

*wClass*

Specifies the logical service class, value is:

WFS\_SERVICE\_CLASS\_SIU

*fwType*

Specifies the type of sensors and indicators supported by this device as a combination of the following flags:

Value	Meaning
WFS_SIU_SENSORS	The device supports input Sensors.
WFS_SIU_DOORS	The device support Door sensors.
WFS_SIU_INDICATORS	The device supports Status Indicators.
WFS_SIU_AUXILIARIES	The device supports Auxiliary Indicators.
WFS_SIU_GUIDLIGHTS	The device supports Guidance Lights.

*fwSensors [...]*

Specifies which Sensors that are available, and if so, which states they can take. A number of sensor types are defined below. Vendor specific sensors are defined starting from the end of the array. The maximum sensor index is WFS\_SIU\_SENSORS\_MAX.

*fwSensors [WFS\_SIU\_OPERATORSWITCH]*

Specifies whether the Operator switch is available, and if so, which states it can take.

Specified as WFS\_SIU\_NOT\_AVAILABLE or as a combination of the following flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	There is no Operator Switch available.	A
WFS_SIU_RUN	The switch can be set in Run mode.	B
WFS_SIU_MAINTENANCE	The switch can be set in Maintenance mode.	B
WFS_SIU_SUPERVISOR	The switch can be set in Supervisors mode.	B

*fwSensors [WFS\_SIU\_TAMPER]*

Specifies whether the Tamper Sensor for the terminal is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Tamper Sensor available.
WFS_SIU_AVAILABLE	The Tamper Sensor is available.

*fwSensors [WFS\_SIU\_INTTAMPER]*

Specifies whether the Tamper Sensor for internal alarm is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no internal Tamper Sensor available.
WFS_SIU_AVAILABLE	The internal Tamper Sensor is available.

*fwSensors [WFS\_SIU\_SEISMIC]*

Specifies whether the Seismic Sensor is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Seismic Sensor available.
WFS_SIU_AVAILABLE	The Seismic Sensor is available.

*fwSensors [WFS\_SIU\_HEAT]*

Specifies whether the Heat Sensor is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Heat Sensor available.
WFS_SIU_AVAILABLE	The Heat Sensor is available.

*fwSensors [WFS\_SIU\_PROXIMITY]*

Specifies whether the Proximity Sensor is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Proximity Sensor available.
WFS_SIU_AVAILABLE	The Proximity Sensor is available.

*fwSensors [WFS\_SIU\_AMBLIGHT]*

Specifies whether the Ambient Light Sensor is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Ambient Light Sensor available.
WFS_SIU_AVAILABLE	The Ambient Light Sensor is available.

*fwDoors [...]*

Specifies which Doors that are available, and if so, which states they can take. A number of door types are defined below. Vendor specific doors are defined starting from the end of the array. The maximum door index is WFS\_SIU\_DOORS\_MAX.

*fwDoors [WFS\_SIU\_CABINET]*

Specifies whether the Cabinet Doors are available, and if so, which states they can take.

Specified as WFS\_SIU\_NOT\_AVAILABLE or as a combination of the following flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	There is no Cabinet Door available.	A
WFS_SIU_LOCKED	The Cabinet Doors can be locked.	B
WFS_SIU_BOLTED	The Cabinet Doors can be bolted.	B
WFS_SIU_CLOSED	The Cabinet Doors can be closed.	B
WFS_SIU_OPEN	The Cabinet Doors can be open.	B

*fwDoors [WFS\_SIU\_SAFE]*

Specifies whether the Safe Doors are available, and if so, which states they can take. Specified as WFS\_SIU\_NOT\_AVAILABLE or as a combination of the following flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	There is no Safe Door available.	A
WFS_SIU_LOCKED	The Safe Doors can be locked.	B
WFS_SIU_BOLTED	The Safe Doors can be bolted.	B
WFS_SIU_CLOSED	The Safe Doors can be closed.	B
WFS_SIU_OPEN	The Safe Doors can be open.	B

*fwDoors [WFS\_SIU\_VANDALSHIELD]*

Specifies whether the Vandal Shield is available, and if so, which states it can take. Specified as WFS\_SIU\_NOT\_AVAILABLE or as a combination of the following flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	There is no Vandal Shield available.	A
WFS_SIU_OPEN	The Vandal Shield can be open.	B
WFS_SIU_SERVICE	The Vandal Shield can be in service position.	B
WFS_SIU_KEYBOARD	The Vandal Shield can be in position that permits access to the keyboard.	B
WFS_SIU_CLOSED	The Vandal Shield can be closed.	B
WFS_SIU_LOCKED	The Vandal Shield can be locked.	B

*fwIndicators [...]*

Specifies which Status Indicators that are available, and if so, which states they can take. A number of Status Indicator types are defined below. Vendor specific indicators are defined starting from the end of the array. The maximum indicator index is WFS\_SIU\_INDICATORS\_MAX.

*fwIndicators [WFS\_SIU\_OPENCLOSE]*

Specifies whether the Open/Closed Indicator is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Open/Closed Indicator available.
WFS_SIU_AVAILABLE	The Open/Closed Indicator is available.

*fwIndicators [WFS\_SIU\_FASCIALIGHT]*

Specifies whether the Fascia Light is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Fascia Light available.
WFS_SIU_AVAILABLE	The Fascia Light is available.

*fwIndicators [WFS\_SIU\_AUDIO]*

Specifies whether the Audio Indicator device is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Audio Indicator available.
WFS_SIU_AVAILABLE	The Audio Indicator is available.

*fwIndicators [WFS\_SIU\_HEATING]*

Specifies whether the internal Heating device is available. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Heating device available.
WFS_SIU_AVAILABLE	The Heating device is available.

*fwAuxiliaries [...]*

Specifies which Auxiliaries that are available, and if so, which states they can take. A number of Auxiliary indicator types are defined below. Vendor specific Auxiliaries are defined starting from the end of the array. The maximum auxiliary index is WFS\_SIU\_AUXILIARIES\_MAX.

*fwAuxiliaries [WFS\_SIU\_VOLUME]*

Specifies if the volume control is available, and if so, the increment/decrement value recommended by the vendor.

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no volume control available.
1, ..., 1000	The recommended increment/decrement value for the volume control.

*fwAuxiliaries [WFS\_SIU\_UPS]*

Specifies if the UPS device is available, and if so, which states it can take. Specified as WFS\_SIU\_NOT\_AVAILABLE or as a combination of the following flags of type B:

Value	Meaning	Type
WFS_SIU_NOT_AVAILABLE	There is no UPS available.	A
WFS_SIU_AVAILABLE	The UPS is available	B
WFS_SIU_LOW	The UPS can indicate that its charge level is low	B
WFS_SIU_ENGAGED	The UPS can be engaged and disengaged by the application	B
WFS_SIU_POWERING	The UPS can indicate that it is powering the system while the main power supply is off.	B
WFS_SIU_RECOVERED	The UPS can indicate that it was engaged when the main power went off	B

*fwGuidLights [...]*

Specifies which Guidance Lights that are available, and if so, which states they can take. A number of guidance light types are defined below. Vendor specific guidance lights are defined starting from the end of the array. The maximum guidance light index is WFS\_SIU\_GUIDLIGHTS\_MAX. The elements of this array are specified as one of the following flags:

Value	Meaning
WFS_SIU_NOT_AVAILABLE	There is no Guidance Light available at this position.
WFS_SIU_AVAILABLE	A Guidance Light is available at this position.

*fwGuidLights [WFS\_SIU\_CARDUNIT]*

Specifies whether the Guidance Light Indicator on the Card Unit (IDC) is available.

*fwGuidLights [WFS\_SIU\_PINPAD]*

Specifies whether the Guidance Light Indicator on the PIN pad unit is available.

*fwGuidLights [WFS\_SIU\_NOTESDISPENSER]*

Specifies whether the Guidance Light Indicator on the note dispenser unit is available.

*fwGuidLights [WFS\_SIU\_COINDISPENSER]*

Specifies whether the Guidance Light Indicator on the coin dispenser unit is available.

*fwGuidLights [WFS\_SIU\_RECEIPTPRINTER]*

Specifies whether the Guidance Light Indicator on the receipt printer unit is available.

*fwGuidLights [WFS\_SIU\_PASSBOOKPRINTER]*

Specifies whether the Guidance Light Indicator on the passbook printer unit is available.

*fwGuidLights [WFS\_SIU\_ENVDEPOSITORY]*

Specifies whether the Guidance Light Indicator on the envelope depository unit is available.

*fwGuidLights [WFS\_SIU\_CHEQUEUNIT]*

Specifies whether the Guidance Light Indicator on the cheque processing unit is available.

*fwGuidLights [WFS\_SIU\_BILLACCEPTOR]*

Specifies whether the Guidance Light Indicator on the bill acceptor unit is available.

*fwGuidLights [WFS\_SIU\_ENVDISPENSER]*

Specifies whether the Guidance Light Indicator on the envelope dispenser unit is available.

*lpszExtra*

Specifies 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 will be null-terminated, with the final string terminating with two null characters.

**Error Codes** There are no additional error codes generated by this command.

**Comments** Applications which require or expect specific information to be present in the *lpszExtra* parameter may not be device or vendor-independent.



## 4. Execute Commands

---

### 4.1 WFS\_CMD\_SIU\_ENABLE\_EVENTS

---

**Description** This command is used to enable or disable events from the Sensors and Indicators Unit. The default condition is that all events are disabled.

**Input Param** LPWFSSIUENABLE lpEnable;

```
typedef struct _wfs_siu_enable
{
    WORD fwSensors [WFS_SIU_SENSORS_SIZE];
    WORD fwDoors [WFS_SIU_DOORS_SIZE];
    WORD fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR lpszExtra;
} WFS_SIU_ENABLE, * LPWFSSIUENABLE;
```

*fwSensors [...]*

Specifies which of the Sensors that should report changes. A number of sensor types are defined below. Vendor specific sensors are defined starting from the end of the array. The maximum sensor index is WFS\_SIU\_SENSORS\_MAX.

*fwSensors [WFS\_SIU\_OPERATORSWITCH]*

Specifies whether the Operator Switch should report whenever the switch changes the operating mode. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Operators Switch should report whenever it changes mode from Run to Maintenance or Supervisor mode or vice versa.
WFS_SIU_DISABLE_EVENT	The Operators Switch should <b>not</b> report any changes of it operating mode.

*fwSensors [WFS\_SIU\_TAMPER]*

Specifies whether the Tamper Sensor should report whenever someone tampers with the terminal. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Tamper Sensor should report whenever it detects any tampering attempt.
WFS_SIU_DISABLE_EVENT	The Tamper Sensor should <b>not</b> report any changes of its status.

*fwSensors [WFS\_SIU\_INTTAMPER]*

Specifies whether the internal Tamper Sensor should report whenever someone tampers with the internal alarm. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Tamper Sensor should report whenever it detects any tampering attempt.
WFS_SIU_DISABLE_EVENT	The Tamper Sensor should <b>not</b> report any changes of its status.

*fwSensors [WFS\_SIU\_SEISMIC]*

Specifies whether the Seismic Sensor should report whenever any seismic activity is detected. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Seismic Sensor should report whenever it detects any seismic activity.
WFS_SIU_DISABLE_EVENT	The Seismic Sensor should <b>not</b> report any changes of its status.

*fwSensors [WFS\_SIU\_HEAT]*

Specifies whether the Heat Sensor should report whenever any excessive heat is detected. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Heat Sensor should report whenever it detects any excessive heat.
WFS_SIU_DISABLE_EVENT	The Heat Sensor should <b>not</b> report any changes of its status.

*fwSensors [WFS\_SIU\_PROXIMITY]*

Specifies whether the Proximity Sensor should report whenever any movement is detected close to the terminal. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Proximity Sensor should report whenever it detects any movement.
WFS_SIU_DISABLE_EVENT	The Proximity Sensor should <b>not</b> report any changes of its status.

*fwSensors [WFS\_SIU\_AMBLIGHT]*

Specifies whether the Ambient Light Sensor should report whenever it detects changes in the ambient light. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Ambient Light Sensor should report whenever it detects a change.
WFS_SIU_DISABLE_EVENT	The Ambient Light Sensor should <b>not</b> report any change.

*fwDoors [...]*

Specifies which of the Doors that should report changes. A number of door types are defined below. Vendor specific doors are defined starting from the end of the array. The maximum door index is WFS\_SIU\_DOORS\_MAX.

*fwDoors [WFS\_SIU\_CABINET]*

Specifies whether the Cabinet Doors sensor should report whenever the doors are opened, closed, bolted or locked. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Cabinet Doors should report whenever the doors are opened, closed, locked or bolted.
WFS_SIU_DISABLE_EVENT	The Cabinet Doors sensor should <b>not</b> report any changes of the doors status.

*fwDoors [WFS\_SIU\_SAFE]*

Specifies whether the Safe Doors should report whenever the doors are opened, closed, bolted or locked. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Safe Doors should report whenever the doors are opened, closed, locked or bolted.
WFS_SIU_DISABLE_EVENT	The Safe Doors should <b>not</b> report any changes of the doors status.

*fwDoors [WFS\_SIU\_VANDALSHIELD]*

Specifies whether the Vandal Shield should report whenever the shield changed position. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Vandal Shield should report whenever the doors are opened or closed.
WFS_SIU_DISABLE_EVENT	The Vandal Shield should <b>not</b> report any changes of the status.

*fwIndicators [...]*

Specifies which of the Status Indicators that should report changes. A number of Status Indicator types are defined below. Vendor specific indicators are defined starting from the end of the array. The maximum indicator index is WFS\_SIU\_INDICATORS\_MAX.

*fwIndicators [WFS\_SIU\_OPENCLOSE]*

Specifies whether the Open/Closed Indicator should report whenever it is turned on (set to open) or turned off (set to closed). Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Open/Closed Indicator should report whenever it is turned on or off.
WFS_SIU_DISABLE_EVENT	The Open/Closed Indicator should <b>not</b> report any changes of the indicator.

*fwIndicators [WFS\_SIU\_FASCIALIGHT]*

Specifies whether the Fascia Light should report whenever it is turned on or turned off.  
Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Fascia Light should report whenever it is turned on or off.
WFS_SIU_DISABLE_EVENT	The Fascia Light should <b>not</b> report any changes.

*fwIndicators [WFS\_SIU\_AUDIO]*

Specifies whether the Audio Indicator should report whenever it is turned on or turned off.  
Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Audio Indicator should report whenever it is turned on or off.
WFS_SIU_DISABLE_EVENT	The Audio Indicator should <b>not</b> report any changes.

*fwIndicators [WFS\_SIU\_HEATING]*

Specifies whether the Heating device should report whenever it is turned on or turned off.  
Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Heating device should report whenever it is turned on or off.
WFS_SIU_DISABLE_EVENT	The Heating device should <b>not</b> report any changes.

*fwAuxiliaries [...]*

Specifies which of the Auxiliary Indicators that should report changes. A number of Auxiliary Indicator types are defined below. Vendor specific indicators are defined starting from the end of the array. The maximum indicator index is WFS\_SIU\_AUXILIARIES\_MAX.

*fwAuxiliaries[WFS\_SIU\_VOLUME]*

Specifies whether the Volume control device should report whenever it is changed or not.  
Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The Volume control device should report whenever it is changed.
WFS_SIU_DISABLE_EVENT	The Volume control device should <b>not</b> report any changes.

*fwAuxiliaries[WFS\_SIU\_UPS]*

Specifies whether the UPS device should report whenever it is changed or not. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.
WFS_SIU_ENABLE_EVENT	The UPS device should report whenever it is changed.
WFS_SIU_DISABLE_EVENT	The UPS device should <b>not</b> report any changes.

*fwGuidLights [...]*

Specifies which of the Guidance Light Indicators should report whenever any of them changes its state. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current reporting status.

WFS_SIU_ENABLE_EVENT	The Light Indicators should report whenever any of them changes its state.
WFS_SIU_DISABLE_EVENT	The Light Indicators should <b>not</b> report any changes of their states.

*fwGuidLights [WFS\_SIU\_CARDUNIT]*

Specifies whether the Guidance Light Indicator on the Card Unit (IDC) should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_PINPAD]*

Specifies whether the Guidance Light Indicator on the PIN pad unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_NOTESDISPENSER]*

Specifies whether the Guidance Light Indicator on the note dispenser unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_COINDISPENSER]*

Specifies whether the Guidance Light Indicator on the coin dispenser unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_RECEIPTPRINTER]*

Specifies whether the Guidance Light Indicator on the receipt printer unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_PASSBOOKPRINTER]*

Specifies whether the Guidance Light Indicator on the passbook printer unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_ENVDEPOSITORY]*

Specifies whether the Guidance Light Indicator on the envelope depository unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_CHEQUEUNIT]*

Specifies whether the Guidance Light Indicator on the cheque processing unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_BILLACCEPTOR]*

Specifies whether the Guidance Light Indicator on the bill acceptor unit should report whenever it changes status.

*fwGuidLights [WFS\_SIU\_ENVDISPENSER]*

Specifies whether the Guidance Light Indicator on the envelope dispenser unit should report whenever it changes status.

*lpszExtra*

Specifies a list of vendor-specific, or any other extended, information. The information is passed as a series of "key=value" strings so that it is easily extensible by service providers. Each string will be null-terminated, with the final string terminating with two null characters.

**Output Param** None.

**Error Codes** The following additional error codes can be generated by this command:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to enable or disable events to a port was invalid because the port does not exist.
WFS_ERR_SIU_SYNTAX	The command was invoked with incorrect input data. E.g. an attempt to both enable <b>and</b> disable events to the same port was made.

**Events** The following additional events can be generated by this command:

Value	Meaning
WFS_EXEE_SIU_PORT_ERROR	A error occurred while enabling or disabling events on one or more ports.

**Comments** No action has been taken if this command returns an error. If a hardware error occurs while executing the command, the command will return OK, but execute event(s) will be generated which indicate(s) the port(s) which have failed.

## 4.2 WFS\_CMD\_SIU\_SET\_PORTS

**Description** This command is used to set or clear one or more output ports (indicators) in the Sensors and Indicators Unit.

**Input Param** LPWFSSIUSETPORTS lpSetPorts;

```
typedef struct _wfs_siu_set_ports
{
    WORD          fwDoors [WFS_SIU_DOORS_SIZE];
    WORD          fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD          fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD          fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR         lpzExtra;
} WFS_SIUSETPORTS, * LPWFSSIUSETPORTS;
```

*fwDoors [WFS\_SIU\_CABINET]*

Specifies whether the Cabinet Doors should be bolted or unbolted. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the Cabinet doors.
WFS_SIU_BOLT	The Cabinet doors are bolted.
WFS_SIU_UNBOLT	The Cabinet doors are unbolted.

*fwDoors [WFS\_SIU\_SAFE]*

Specifies whether the Safe Doors should be bolted or unbolted. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the Safe doors.
WFS_SIU_BOLT	The Safe doors are bolted.
WFS_SIU_UNBOLT	The Safe doors are unbolted.

*fwDoors [WFS\_SIU\_VANDALSHIELD]*

Specifies whether the Vandal Shield should change position. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current position of the Vandal shield.
WFS_SIU_OPEN	The Vandal Shield is opened.
WFS_SIU_SERVICE	The Vandal Shield is set in service position.
WFS_SIU_KEYBOARD	The Vandal Shield is set in position that permits access to the keyboard.
WFS_SIU_CLOSED	The Vandal Shield is closed.

*fwIndicators [WFS\_SIU\_OPENCLOSE]*

Specifies whether the Open/Closed Indicator should show Open or Close to a consumer. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the indicator.
WFS_SIU_CLOSED	The indicator is changed to show that the terminal is closed for a consumer.
WFS_SIU_OPEN	The indicator is changed to show that the terminal is open to be used by a consumer.

*fwIndicators [WFS\_SIU\_FASCIALIGHT]*

Specifies whether the Fascia Lights should be turned on or off. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the light.
WFS_SIU_OFF	The Fascia Light is turned off.
WFS_SIU_ON	The Fascia Light is turned on.

*fwIndicators [WFS\_SIU\_AUDIO]*

Specifies whether the Audio Indicator should be turned on or off. Specified as one of the following flags of type A and B, or as WFS\_SIU\_CONTINUOUS in combination with one of the flags of type B:

Value	Meaning	Type
WFS_SIU_NO_CHANGE	Do not change the current status of the beeper.	A
WFS_SIU_OFF	The Audio Indicator is turned off.	A
WFS_SIU_KEYPRESS	The Audio Indicator sounds a key click signal.	B
WFS_SIU_EXCLAMATION	The Audio Indicator sounds a exclamation signal.	B
WFS_SIU_WARNING	The Audio Indicator sounds a warning signal.	B
WFS_SIU_ERROR	The Audio Indicator sounds a error signal.	B
WFS_SIU_CRITICAL	The Audio Indicator sounds a critical error signal.	B
WFS_SIU_CONTINUOUS	The Audio Indicator sound is turned on continuously.	C

*fwIndicators [WFS\_SIU\_HEATING]*

Specifies whether the internal Heating device should be turned on or off. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the light.
WFS_SIU_OFF	The Heating is turned off.
WFS_SIU_ON	The Heating is turned on.

*fwAuxiliaries [WFS\_SIU\_VOLUME]*

Specifies whether the value of the volume control should be changed or not. If so, the value of volume control is defined in an interval from 1 to 1000 were 1 is the lowest volume level and 1000 is the highest volume level. Specified as one of the following values:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the light.
1, ..., 1000	The volume level. This field is handled as an unsigned short. If a value greater than 1000 is used, the provider will map the value to 1000.

*fwAuxiliaries [WFS\_SIU\_UPS]*

Specifies whether the UPS device should be engaged or disengaged. The UPS device should not be engaged when the charge level is low. Specified as one of the following flags:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the UPS device.
WFS_SIU_ENGAGE	Engage the UPS.

WFS\_SIU\_DISENGAGE Disengage the UPS.

*fwGuidLights [...]*

Specifies whether the Guidance Light Indicators should be turned on or off, or if they should flash. All member elements of the Guidance Lights structure can be specified as one of the following values:

Value	Meaning
WFS_SIU_NO_CHANGE	Do not change the current status of the Light Indicator.
WFS_SIU_OFF	The Light Indicator is turned off.
WFS_SIU_SLOW_FLASH	The Light Indicator is set to flash slowly.
WFS_SIU_MEDIUM_FLASH	The light is blinking medium frequency.
WFS_SIU_QUICK_FLASH	The Light Indicator is set to flash quickly.
WFS_SIU_CONTINUOUS	The Light Indicator is turned on continuously (steady).

*fwGuidLights [WFS\_SIU\_CARDUNIT]*

Specifies the state of the Guidance Light Indicator on the Card Unit (IDC).

*fwGuidLights [WFS\_SIU\_PINPAD]*

Specifies the state of the Guidance Light Indicator on the PIN pad unit.

*fwGuidLights [WFS\_SIU\_NOTESDISPENSER]*

Specifies the state of the Guidance Light Indicator on the note dispenser unit.

*fwGuidLights [WFS\_SIU\_COINDISPENSER]*

Specifies the state of the Guidance Light Indicator on the coin dispenser unit.

*fwGuidLights [WFS\_SIU\_RECEIPTPRINTER]*

Specifies the state of the Guidance Light Indicator on the receipt printer unit.

*fwGuidLights [WFS\_SIU\_PASSBOOKPRINTER]*

Specifies the state of the Guidance Light Indicator on the passbook printer unit.

*fwGuidLights [WFS\_SIU\_ENVDEPOSITORY]*

Specifies the state of the Guidance Light Indicator on the envelope depository unit.

*fwGuidLights [WFS\_SIU\_CHEQUEUNIT]*

Specifies the state of the Guidance Light Indicator on the cheque processing unit.

*fwGuidLights [WFS\_SIU\_BILLACCEPTOR]*

Specifies the state of the Guidance Light Indicator on the bill acceptor unit.

*fwGuidLights [WFS\_SIU\_ENVDISPENSER]*

Specifies the state of the Guidance Light Indicator on the envelope dispenser unit.

*lpszExtra*

Specifies a list of vendor-specific, or any other extended, information. The information is passed as a series of "key=value" strings so that it is easily extensible by service providers. Each string will be null-terminated, with the final string terminating with two null characters.

**Output Param** None.



**Error Codes**      The following additional error codes can be generated by this command:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to set a port to a new value was invalid because the port does not exist or the port is pre-configured as an input port.
WFS_ERR_SIU_SYNTAX	The command was invoked with incorrect input data.

**Events**      The following additional events can be generated by this command:

Value	Meaning
WFS_EXEE_SIU_PORT_ERROR	An error occurred while attempting to set or clear one or more output ports (indicators).

**Comments**      No action has been taken if this command returns an error. If a hardware error occurs while executing the command, the command will return OK, but execute event(s) will be generated which indicate(s) the port(s) which have failed.

### 4.3 WFS\_CMD\_SIU\_SET\_DOOR

**Description**      This command is used to set the status of one of the Doors.

**Input Param**      LPWFSSIUSETDOOR      lpSetDoor;

```
typedef struct _wfs_siu_set_door
{
    WORD            wDoor;
    WORD            fwCommand;
} WFSSETDOOR, * LPWFSSIUSETDOOR;
```

*wDoor*

Specifies the index of the Door to set as one of the following values:

Value	Meaning
WFS_SIU_CABINET	Bolt/unbolt the Cabinet doors.
WFS_SIU_SAFE	Bolt/unbolt the Safe doors.
WFS_SIU_VANDALSHIELD	Set position of the Vandal Shield.

*fwCommand*

Specifies if the Cabinet or Safe doors should be bolted or unbolted or if the position of the Vandal Shield should be changed, as one of the following flags:

Value	Meaning
WFS_SIU_BOLT	Bolt the Safe- or Cabinet doors.
WFS_SIU_UNBOLT	Unbolt the Safe- or Cabinet doors.
WFS_SIU_OPEN	Open the Vandal Shield.
WFS_SIU_SERVICE	Position the Vandal Shield in service position.
WFS_SIU_KEYBOARD	Position the Vandal Shield to permit access to the keyboard.
WFS_SIU_CLOSED	Close the Vandal Shield.

See **WFS\_CMD\_SIU\_SET\_PORTS** command for a detailed description.

**Output Param**      None.

**Error Codes**      The following additional error codes can be generated by this command:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to set a port to a new value was invalid because the port does not exist or the port is pre-configured as an input port.
WFS_ERR_SIU_SYNTAX	The command was invoked with incorrect input data.
WFS_ERR_SIU_PORT_ERROR	A hardware error occurred while executing the command.

**Events** The following additional events can be generated by this command:

Value	Meaning
WFS_EXEE_SIU_PORT_ERROR	An error occurred while attempting to set the status of the door.

**Comments** None.

#### 4.4 WFS\_CMD\_SIU\_SET\_INDICATOR

**Description** This command is used to set the status of an Indicator.

**Input Param** LPWFSSIUSETINDICATOR lpSetIndicator;

```
typedef struct _wfs_siu_set_indicator
{
    WORD        wIndicator;
    WORD        fwCommand;
} WFSIUSETINDICATOR, * LPWFSSIUSETINDICATOR;
```

##### *wIndicator*

Specifies the index of the Status Indicator to set as one of the following values:

Value	Meaning
WFS_SIU_OPENCLOSE	Set Open/Close indicator for the consumer.
WFS_SIU_FASCIALIGHT	Turn on/off the fascia light.
WFS_SIU_AUDIO	Turn on/off the audio indicator.
WFS_SIU_HEATING	Turn on/off the heating device.

##### *fwCommand*

Specifies the commands for the open/close indicator, fascia light, audio indicator and heating device, as one of the following flags:

Value	Meaning
WFS_SIU_CLOSED	The Open/Close indicator is changed to show that the terminal is closed for a consumer.
WFS_SIU_OPEN	The Open/Close indicator is changed to show that the terminal is open to be used by a consumer.
WFS_SIU_OFF	The Audio Indicator, Fascia Light or Heating is turned off.
WFS_SIU_ON	The Fascia Light or Heating is turned on.
WFS_SIU_KEYPRESS	The Audio Indicator sounds a key click signal.
WFS_SIU_EXCLAMATION	The Audio Indicator sounds a exclamation signal.
WFS_SIU_WARNING	The Audio Indicator sounds a warning signal.
WFS_SIU_ERROR	The Audio Indicator sounds a error signal.
WFS_SIU_CRITICAL	The Audio Indicator sounds a critical error signal.
WFS_SIU_CONTINUOUS	The Audio Indicator sound is turned on continuously.

See **WFS\_CMD\_SIU\_SET\_PORTS** command for a detailed description.

**Output Param** None.

**Error Codes** The following additional error codes can be generated by this command:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to set a port to a new value was invalid because the port does not exist or the port is pre-configured as an input port.
WFS_ERR_SIU_SYNTAX	The command was invoked with incorrect input data.
WFS_ERR_SIU_PORT_ERROR	A hardware error occurred while executing the command.

<b>Events</b>	The following additional events can be generated by this command:				
	<table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>WFS_EXEE_SIU_PORT_ERROR</td> <td>An error occurred while attempting to set the status of the indicator.</td> </tr> </tbody> </table>	Value	Meaning	WFS_EXEE_SIU_PORT_ERROR	An error occurred while attempting to set the status of the indicator.
Value	Meaning				
WFS_EXEE_SIU_PORT_ERROR	An error occurred while attempting to set the status of the indicator.				

**Comments** None.

## 4.5 WFS\_CMD\_SIU\_SET\_AUXILIARY

**Description** This command is used to set the status of an Auxiliary indicator.

**Input Param** LPWFSSIUSETAUXILIARY lpSetAuxiliary;

```
typedef struct _wfs_siu_set_auxiliary
{
    WORD          wAuxiliary;
    WORD          fwCommand;
} WFSIUSETAUXILIARY, * LPWFSSIUSETAUXILIARY;
```

### *wAuxiliary*

Specifies the index of the Auxiliary indicator to set as one of the following values:

Value	Meaning
WFS_SIU_VOLUME	Set the value of the volume control.
WFS_SIU_UPS	Set the value of the UPS.

### *fwCommand*

It specifies the values for the volume control or the command to the UPS device. Specified as one of the following values:

Value	Meaning
1, ..., 1000	The volume level. This field is handled as an unsigned short. If a value greater than 1000 is used, the provider will map the value to 1000.
WFS_SIU_ENGAGE	Engage the UPS.
WFS_SIU_DISENGAGE	Disengage the UPS.

See **WFS\_CMD\_SIU\_SET\_PORTS** command for a detailed description.

**Output Param** None.

**Error Codes** The following additional error codes can be generated by this command:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to set a port to a new value was invalid because the port does not exist or the port is pre-configured as an input port.
WFS_ERR_SIU_SYNTAX	The command was invoked with incorrect input data.
WFS_ERR_SIU_PORT_ERROR	A hardware error occurred while executing the command.

**Events** The following additional events can be generated by this command:

Value	Meaning
WFS_EXEE_SIU_PORT_ERROR	An error occurred while attempting to set the status of the auxiliary indicator.

**Comments** None.

## 4.6 WFS\_CMD\_SIU\_SET\_GUIDLIGHT

---

**Description** This command is used to set the status of a Guidance Light.

**Input Param** LPWFSSIUSETGUIDLIGHT lpSetGuidLight;

```
typedef struct _wfs_siu_set_guidlight
{
    WORD          wGuidLight;
    WORD          fwCommand;
} WFSIUSETGUIDLIGHT, * LPWFSSIUSETGUIDLIGHT;
```

### *wGuidLights*

Specifies the index of the Guidance Light to set as one of the following values:

Value	Meaning
WFS_SIU_CARDUNIT	Set the state of the Guidance Light Indicator on the Card Unit (IDC).
WFS_SIU_PINPAD	Set the state of the Guidance Light Indicator on the PIN pad unit.
WFS_SIU_NOTESDISPENSER	Set the state of the Guidance Light Indicator on the note dispenser unit.
WFS_SIU_COINDISPENSER	Set the state of the Guidance Light Indicator on the coin dispenser unit.
WFS_SIU_RECEIPTPRINTER	Set the state of the Guidance Light Indicator on the receipt printer unit.
WFS_SIU_PASSBOOKPRINTER	Set the state of the Guidance Light Indicator on the passbook printer unit.
WFS_SIU_ENVDEPOSITORY	Set the state of the Guidance Light Indicator on the envelope depository unit.
WFS_SIU_CHEQUEUNIT	Set the state of the Guidance Light Indicator on the cheque processing unit.
WFS_SIU_BILLACCEPTOR	Set the state of the Guidance Light Indicator on the bill acceptor unit.
WFS_SIU_ENVDISPENSER	Set the state of the Guidance Light Indicator on the envelope dispenser unit.

### *fwCommand*

Specifies the state of the Guidance Light indicators, as one of the following flags:

Value	Meaning
WFS_SIU_OFF	The Light Indicator is turned off.
WFS_SIU_SLOW_FLASH	The Light Indicator is set to flash slowly.
WFS_SIU_MEDIUM_FLASH	The light is blinking medium frequency.
WFS_SIU_QUICK_FLASH	The Light Indicator is set to flash quickly.
WFS_SIU_CONTINUOUS	The Light Indicator is turned on continuously (steady).

See **WFS\_CMD\_SIU\_SET\_PORTS** command for a detailed description.

**Output Param** None.

**Error Codes** The following additional error codes can be generated by this command:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to set a port to a new value was invalid because the port does not exist or the port is pre-configured as an input port.
WFS_ERR_SIU_SYNTAX	The command was invoked with incorrect input data.

**Events** The following additional events can be generated by this command:

Value	Meaning
WFS_EXEE_SIU_PORT_ERROR	An error occurred while attempting to set or clear one or more output ports (indicators).

Comments None.

## 5. Events

---

### 5.1 WFS\_SRVE\_SIU\_PORT\_STATUS

---

**Description** This event id is used to specify that a port has changed its state, due to the result of a command or to some external condition. Reporting of this event is controlled by the WFS\_CMD\_SIU\_ENABLE\_EVENTS command. Event reporting is disabled as a default situation.

**Event Param** LPWFSSIUPORTEVENT lpPortEvent;

```
typedef struct _wfs_siu_port_event
{
    WORD          wPortType;
    WORD          wPortIndex;
    WORD          wPortStatus;
    LPSTR         lpszExtra;
} WFS_SIUPORTEVENT, * LPWFSSIUPORTEVENT;
```

*wPortType*

Specifies the type of sensors and indicators that has changed state by one of the following flags:

Value	Meaning
WFS_SIU_SENSORS	A port in the input Sensors has changed state.
WFS_SIU_DOORS	A port in the Door sensors has changed state.
WFS_SIU_INDICATORS	A port in the Status Indicators has changed state.
WFS_SIU_AUXILIARIES	A port in the Auxiliary Indicators has changed state.
WFS_SIU_GUIDLIGHTS	A port in the Guidance Lights has changed state.

*wPortIndex*

Specifies the index of the port that has changed state by one of the following values:

Value	Meaning
WFS_SIU_OPERATORSWITCH	The Operator Switch has changed its state.
WFS_SIU_TAMPER	The Tamper Sensor has changed its state.
WFS_SIU_INTTAMPER	The internal Tamper Sensor has changed its state.
WFS_SIU_SEISMIC	The Seismic Sensor has changed its state.
WFS_SIU_HEAT	The Heat Sensor has changed its state.
WFS_SIU_PROXIMITY	The Proximity Sensor has changed its state.
WFS_SIU_AMBLIGHT	The Ambient Light Sensor has changed its state.
WFS_SIU_CABINET	The Cabinet Doors have changed their state.
WFS_SIU_SAFE	The Safe Doors have changed their state.
WFS_SIU_VANDALSHIELD	The Vandal Shield has changed its state.
WFS_SIU_OPENCLOSE	The Open/Close Indicator state has changed.
WFS_SIU_FASCIALIGHT	The Fascia Light state has changed.
WFS_SIU_AUDIO	The Audio Indicator state has changed.
WFS_SIU_HEATING	The Heating device state has changed.
WFS_SIU_VOLUME	The Volume control device has changed its value.
WFS_SIU_UPS	The UPS device state has changed.
WFS_SIU_CARDUNIT	The Guidance Light state for the card unit has changed.

WFS_SIU_PINPAD	The Guidance Light state for the PIN pad unit has changed.
WFS_SIU_NOTESDISPENSER	The Guidance Light state for the note dispenser unit has changed.
WFS_SIU_COINDISPENSER	The Guidance Light state for the coin dispenser unit has changed.
WFS_SIU_RECEIPTPRINTER	The Guidance Light state for the receipt printer unit has changed.
WFS_SIU_PASSBOOKPRINTER	The Guidance Light state for the passbook printer unit has changed.
WFS_SIU_ENVDEPOSITORY	The Guidance Light state for the envelope depository unit has changed.
WFS_SIU_CHEQUEUNIT	The Guidance Light state for the cheque unit has changed.
WFS_SIU_BILLACCEPTOR	The Guidance Light state for the bill acceptor unit has changed.
WFS_SIU_ENVDISPENSER	The Guidance Light state for the envelope dispenser unit has changed.

*wPortStatus*

Specifies the new state of the port indicated in the *wPortEvent*. See the WFS\_INF\_SIU\_STATUS information command for the possible values.

*lpzExtra*

Specifies 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 will be null-terminated, with the final string terminating with two null characters.

**Comments**      None.

## 5.2 WFS\_EXEE\_SIU\_PORT\_ERROR

---

**Description**      This event id is used to specify that a port has detected an error.

**Event Param**      LPWFSSSIUPORTERROR    pPortError;

```
typedef struct _wfs_siu_port_error
{
    WORD           wPortType;
    WORD           wPortIndex;
    HRESULT        PortError;
    WORD           wPortStatus;
    LPSTR          lpzExtra;
} WFS_SSIUPORTERROR, * LPWFSSSIUPORTERROR;
```

*wPortType*

Specifies the type of sensors and indicators that has detected an error by one of the following flags:

Value	Meaning
WFS_SIU_SENSORS	A port in the input Sensors has detected an error.
WFS_SIU_DOORS	A port in the Door sensors has detected an error.
WFS_SIU_INDICATORS	A port in the Status Indicators has detected an error.
WFS_SIU_AUXILIARIES	A port in the Auxiliary Indicators has detected an error.
WFS_SIU_GUIDLIGHTS	A port in the Guidance Lights has detected an error.

*wPortIndex*

Specifies the index of the port that has detected an error by one of the following values:

Value	Meaning
WFS_SIU_OPERATORSWITCH	The Operator Switch has detected an error.

WFS_SIU_TAMPER	The Tamper Sensor has detected an error.
WFS_SIU_INTTAMPER	The internal Tamper Sensor has detected an error.
WFS_SIU_SEISMIC	The Seismic Sensor has detected an error.
WFS_SIU_HEAT	The Heat Sensor has detected an error.
WFS_SIU_PROXIMITY	The Proximity Sensor has detected an error.
WFS_SIU_AMBLIGHT	The Ambient Light Sensor has detected an error.
WFS_SIU_CABINET	The Cabinet Doors have detected an error.
WFS_SIU_SAFE	The Safe Doors have detected an error.
WFS_SIU_VANDALSHIELD	The Vandal Shield has detected an error.
WFS_SIU_OPENCLOSE	The Open/Close Indicator has detected an error.
WFS_SIU_FASCIALIGHT	The Fascia Light state has detected an error.
WFS_SIU_AUDIO	The Audio Indicator state has detected an error.
WFS_SIU_HEATING	The Heating device state has detected an error.
WFS_SIU_VOLUME	The Volume control device has detected an error.
WFS_SIU_UPS	The UPS device has detected an error.
WFS_SIU_CARDUNIT	The Guidance Light state for the card unit has detected an error.
WFS_SIU_PINPAD	The Guidance Light state for the PIN pad unit has detected an error.
WFS_SIU_NOTESDISPENSER	The Guidance Light state for the note dispenser unit has detected an error.
WFS_SIU_COINDISPENSER	The Guidance Light state for the coin dispenser unit has detected an error.
WFS_SIU_RECEIPTPRINTER	The Guidance Light state for the receipt printer unit has detected an error.
WFS_SIU_PASSBOOKPRINTER	The Guidance Light state for the passbook printer unit has detected an error.
WFS_SIU_ENVDEPOSITORY	The Guidance Light state for the envelope depository unit has detected an error.
WFS_SIU_CHEQUEUNIT	The Guidance Light state for the cheque unit has detected an error.
WFS_SIU_BILLACCEPTOR	The Guidance Light state for the bill acceptor unit has detected an error.
WFS_SIU_ENVDISPENSER	The Guidance Light state for the envelope dispenser unit has detected an error.

*PortError*

Specifies the error of the port indicated in the *wPortType* and *wPortIndex* by one of the following values:

Value	Meaning
WFS_ERR_SIU_INVALID_PORT	An attempt to enable or disable events to a port was invalid because the port does not exist.
WFS_ERR_SIU_SYNTAX	Syntax error in the input parameters. E.g. an attempt to both enable <b>and</b> disable events to the same port was made.
WFS_ERR_SIU_PORT_ERROR	A hardware error occurred while executing a command.

*wPortStatus*

Specifies the new state of the port indicated in the *wPortEvent*. See the WFS\_INF\_SIU\_STATUS information command for the possible values.

*lpzExtra*

Specifies 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 will be null-terminated, with the final string terminating with two null characters.

**Comments**      None.



## 6. C - header file

```

/*****
*
* xfssiu.h      XFS - definitions
*              for the Sensors and Indicators Unit - services
*
*              Version 2.00 (11/11/96)
*
*****/

#ifndef __INC_XFSSIU__H
#define __INC_XFSSIU__H

#ifdef __cplusplus
extern "C" {
#endif

#include <xfsapi.h>

/* be aware of alignment */
#pragma pack (push, 1)

/* values of WFS_SIUCLASS.wClass */
#define WFS_SERVICE_CLASS_SIU (8)

#define WFS_SERVICE_CLASS_NAME_SIU "SIU"
#define WFS_SERVICE_CLASS_VERSION_SIU 0x0002

#define SIU_SERVICE_OFFSET (WFS_SERVICE_CLASS_SIU * 100)

/* SIU Info Commands */
#define WFS_INF_SIU_STATUS (SIU_SERVICE_OFFSET + 1)
#define WFS_INF_SIU_CAPABILITIES (SIU_SERVICE_OFFSET + 2)

/* SIU Command Verbs */
#define WFS_CMD_SIU_ENABLE_EVENTS (SIU_SERVICE_OFFSET + 1)
#define WFS_CMD_SIU_SET_PORTS (SIU_SERVICE_OFFSET + 2)
#define WFS_CMD_SIU_SET_DOOR (SIU_SERVICE_OFFSET + 3)
#define WFS_CMD_SIU_SET_INDICATOR (SIU_SERVICE_OFFSET + 4)
#define WFS_CMD_SIU_SET_AUXILIARY (SIU_SERVICE_OFFSET + 5)
#define WFS_CMD_SIU_SET_GUIDLIGHT (SIU_SERVICE_OFFSET + 6)

/* SIU Messages */
#define WFS_SRVE_SIU_PORT_STATUS (SIU_SERVICE_OFFSET + 1)
#define WFS_EXEE_SIU_PORT_ERROR (SIU_SERVICE_OFFSET + 2)

/* Values of WFS_SIUSTATUS.fwDevice */
#define WFS_SIU_DEVONLINE WFS_STAT_DEVONLINE
#define WFS_SIU_DEVOFFLINE WFS_STAT_DEVOFFLINE
#define WFS_SIU_DEVPOWEROFF WFS_STAT_DEVPOWEROFF
#define WFS_SIU_DEVBUSY WFS_STAT_DEVBUSY
#define WFS_SIU_DEVNODVICE WFS_STAT_DEVNODVICE
#define WFS_SIU_DEVHWERROR WFS_STAT_DEVHWERROR
#define WFS_SIU_DEVUSERERROR WFS_STAT_DEVUSERERROR

/* Size and max index of fwSensors array */
#define WFS_SIU_SENSORS_SIZE (32)
#define WFS_SIU_SENSORS_MAX (WFS_SIU_SENSORS_SIZE - 1)

/* Size and max index of fwDoors array */
#define WFS_SIU_DOORS_SIZE (16)
#define WFS_SIU_DOORS_MAX (WFS_SIU_DOORS_SIZE - 1)

/* Size and max index of fwIndicators array */
#define WFS_SIU_INDICATORS_SIZE (16)
#define WFS_SIU_INDICATORS_MAX (WFS_SIU_INDICATORS_SIZE - 1)

/* Size max index of fwAuxiliaries array */
#define WFS_SIU_AUXILIARIES_SIZE (16)
#define WFS_SIU_AUXILIARIES_MAX (WFS_SIU_AUXILIARIES_SIZE - 1)

```

```
/* Size and max index of fwGuidLights array */
#define WFS_SIU_GUIDLIGHTS_SIZE (16)
#define WFS_SIU_GUIDLIGHTS_MAX (WFS_SIU_GUIDLIGHTS_SIZE - 1)

/* Indices of WFSSIUSTATUS.fwSensors [...]
   WFSSIUCAPS.fwSensors [...]
   WFSSIUENABLE.fwSensors [...]
   WFSSIUPORTEVENT.wPortIndex
   WFSSIUPORTERROR.wPortIndex */
#define WFS_SIU_OPERATORSWITCH (0)
#define WFS_SIU_TAMPER (1)
#define WFS_SIU_INTTAMPER (2)
#define WFS_SIU_SEISMIC (3)
#define WFS_SIU_HEAT (4)
#define WFS_SIU_PROXIMITY (5)
#define WFS_SIU_AMBLIGHT (6)

/* Indices of WFSSIUSTATUS.fwDoors [...]
   WFSSIUCAPS.fwDoors [...]
   WFSSIUENABLE.fwDoors [...]
   WFSSIUSETPORT.fwDoors [...]
   WFSSIUSETDOORS.wDoor
   WFSSIUPORTEVENT.wPortIndex
   WFSSIUPORTERROR.wPortIndex */
#define WFS_SIU_CABINET (0)
#define WFS_SIU_SAFE (1)
#define WFS_SIU_VANDALSHIELD (2)

/* Indices of WFSSIUSTATUS.fwIndicators [...]
   WFSSIUCAPS.fwIndicators [...]
   WFSSIUENABLE.fwIndicators [...]
   WFSSIUSETPORT.wIndicators [...]
   WFSSIUSETINDICATORS.wIndicator
   WFSSIUPORTEVENT.wPortIndex
   WFSSIUPORTERROR.wPortIndex */
#define WFS_SIU_OPENCLOSE (0)
#define WFS_SIU_FASCIALIGHT (1)
#define WFS_SIU_AUDIO (2)
#define WFS_SIU_HEATING (3)

/* Indices of WFSSIUSTATUS.fwAuxiliaries [...]
   WFSSIUCAPS.fwAuxiliaries [...]
   WFSSIUENABLE.fwAuxiliaries [...]
   WFSSIUSETPORT.wAuxiliaries [...]
   WFSSIUSETAUXILIARIES.wAuxiliary
   WFSSIUPORTEVENT.wPortIndex
   WFSSIUPORTERROR.wPortIndex */
#define WFS_SIU_VOLUME (0)
#define WFS_SIU_UPS (1)

/* Indices of WFSSIUSTATUS.fwGuidLights [...]
   WFSSIUCAPS.fwGuidLights [...]
   WFSSIUENABLE.fwGuidLights [...]
   WFSSIUSETPORT.wGuidLights [...]
   WFSSIUSETGUIDLIGHTS.wGuidLight
   WFSSIUPORTEVENT.wPortIndex
   WFSSIUPORTERROR.wPortIndex */
#define WFS_SIU_CARDUNIT (0)
#define WFS_SIU_PINPAD (1)
#define WFS_SIU_NOTESDISPENSER (2)
#define WFS_SIU_COINDISPENSER (3)
#define WFS_SIU_RECEIPTPRINTER (4)
#define WFS_SIU_PASSBOOKPRINTER (5)
#define WFS_SIU_ENVDEPOSITORY (6)
#define WFS_SIU_CHEQUEUNIT (7)
#define WFS_SIU_BILLACCEPTOR (8)
#define WFS_SIU_ENVDISPENSER (9)

/* Values of WFSSIUSTATUS.fwSensors [...]
   WFSSIUSTATUS.fwDoors [...]
   WFSSIUSTATUS.fwIndicators [...]
   WFSSIUSTATUS.fwAuxiliaries [...]
```

```

WFSSIUSTATUS.fwGuidLights [...]
WFSSIUCAPS.fwSensors [...]
WFSSIUCAPS.fwDoors [...]
WFSSIUCAPS.fwIndicators [...]
WFSSIUCAPS.fwAuxiliaries [...]
WFSSIUCAPS.fwGuidLights [...] */
#define      WFS_SIU_NOT_AVAILABLE          (0x0000)
#define      WFS_SIU_AVAILABLE             (0x0001)

/* Values of WFSSIUSTATUS.fwSensors [WFS_SIU_OPERATORSWITCH]
WFSSIUCAPS.fwSensors [WFS_SIU_OPERATORSWITCH]
WFSSIUPORTEVENT.fwPortStatus
WFSSIUPORTERROR.fwPortStatus */
#define      WFS_SIU_RUN                    (0x0001)
#define      WFS_SIU_MAINTENANCE           (0x0002)
#define      WFS_SIU_SUPERVISOR           (0x0004)

/* Values of WFSSIUSTATUS.fwDoors [...]
WFSSIUSTATUS.fwIndicators [WFS_SIU_OPENCLOSE]
WFSSIUCAPS.fwDoors [...]
WFSSIUCAPS.fwIndicators [WFS_SIU_OPENCLOSE]
WFSSIUSETPORT.fwDoors [...]
WFSSIUSETPORT.fwIndicators [WFS_SIU_OPENCLOSE]
WFSSIUSETDOOR.wDoor
WFSSIUSETINDICATOR.wCommand
WFSSIUPORTEVENT.wPortStatus
WFSSIUPORTERROR.wPortStatus */
#define      WFS_SIU_CLOSED                (0x0001)
#define      WFS_SIU_OPEN                  (0x0002)
#define      WFS_SIU_LOCKED                (0x0004)
#define      WFS_SIU_BOLTED                (0x0008)
#define      WFS_SIU_SERVICE               (0x0010)
#define      WFS_SIU_KEYBOARD              (0x0020)
#define      WFS_SIU_AJAR                  (0x0040)
#define      WFS_SIU_JAMMED                (0x0080)

/* Values of WFSSIUSTATUS.fwIndicators [WFS_SIU_AUDIO]
WFSSIUSETPORT.fwIndicators [WFS_SIU_AUDIO]
WFSSIUSETINDICATOR.wCommand
WFSSIUPORTEVENT.wPortStatus
WFSSIUPORTERROR.wPortStatus */
#define      WFS_SIU_KEYPRESS              (0x0002)
#define      WFS_SIU_EXCLAMATION           (0x0004)
#define      WFS_SIU_WARNING               (0x0008)
#define      WFS_SIU_ERROR                 (0x0010)
#define      WFS_SIU_CRITICAL               (0x0020)

/* Values of WFSSIUSTATUS.fwSensors [...]
WFSSIUSTATUS.fwIndicators [...]
WFSSIUSTATUS.fwGuidLights [...]
WFSSIUCAPS.fwSensors [...]
WFSSIUCAPS.fwIndicators [...]
WFSSIUCAPS.fwGuidLights [...]
WFSSIUSETPORT.fwIndicators [...]
WFSSIUSETPORT.fwGuidLights [...]
WFSSIUSETINDICATORS.fwCommand [...]
WFSSIUSETGUIDLIGHTS.fwCommand [...]
WFSSIUPORTEVENT.wPortStatus
WFSSIUPORTERROR.wPortStatus */
#define      WFS_SIU_OFF                   (0x0001)
#define      WFS_SIU_ON                     (0x0002)
#define      WFS_SIU_SLOW_FLASH            (0x0004)
#define      WFS_SIU_MEDIUM_FLASH          (0x0008)
#define      WFS_SIU_QUICK_FLASH           (0x0010)
#define      WFS_SIU_CONTINUOUS            (0x0080)

/* Values of WFSSIUSTATUS.fwSensors [WFS_SIU_PROXIMITY]
WFSSIUPORTEVENT.wPortStatus
WFSSIUPORTERROR.wPortStatus */
#define      WFS_SIU_PRESENT                (0x0001)
#define      WFS_SIU_NOT_PRESENT            (0x0002)

```

```
/* Values of WFSSIUSTATUS.fwSensors [WFS_SIU_AMBLIGHT]
   WFSSIUCAPS.fwSensors [WFS_SIU_AMBLIGHT]
   WFSSIUORTEVENT.fwPortStatus
   WFSSIUORTERROR.fwPortStatus */
#define WFS_SIU_VERY_DARK (0x0001)
#define WFS_SIU_DARK (0x0002)
#define WFS_SIU_MEDIUM_LIGHT (0x0004)
#define WFS_SIU_LIGHT (0x0008)
#define WFS_SIU_VERY_LIGHT (0x0010)

/* Values of WFSSIUSTATUS.fwAuxiliaries [WFS_SIU_UPS]
   WFSSIUCAPS.fwAuxiliaries [WFS_SIU_UPS]
   WFSSIUORTEVENT.wPortStatus
   WFSSIUORTERROR.wPortStatus */
#define WFS_SIU_LOW (0x0002)
#define WFS_SIU_ENGAGED (0x0004)
#define WFS_SIU_POWERING (0x0008)
#define WFS_SIU_RECOVERED (0x0010)

/* Values of WFSSIUCAPS.fwType */
#define WFS_SIU_SENSORS (0x0001)
#define WFS_SIU_DOORS (0x0002)
#define WFS_SIU_INDICATORS (0x0004)
#define WFS_SIU_AUXILIARIES (0x0008)
#define WFS_SIU_GUIDLIGHTS (0x0010)

/* Values of WFSSIUENABLE.fwSensors [...]
   WFSSIUENABLE.fwDoors [...]
   WFSSIUENABLE.fwIndicators [...]
   WFSSIUENABLE.fwAuxiliaries [...]
   WFSSIUENABLE.fwGuidLights [...]
   WFSSIUSETPORTS.fwDoors [...]
   WFSSIUSETPORTS.fwIndicators [...]
   WFSSIUSETPORTS.fwAuxiliaries [...]
   WFSSIUSETPORTS.fwGuidLights [...] */
#define WFS_SIU_NO_CHANGE (0x0000)
#define WFS_SIU_ENABLE_EVENT (0x0001)
#define WFS_SIU_DISABLE_EVENT (0x0002)

/* Values of WFSSIUSETPORTS.fwDoors [...]
   WFSSIUSETDOORS.fwCommand [...] */
#define WFS_SIU_BOLT (0x0001)
#define WFS_SIU_UNBOLT (0x0002)

/* Values of WFSSIUSETPORTS.fwAuxiliaries [WFS_SIU_UPS]
   WFSSIUSETAUXILIARY.wAuxiliary [WFS_SIU_UPS] */
#define WFS_SIU_ENGAGE (0x0001)
#define WFS_SIU_DISENGAGE (0x0002)

/* XFS SIU Errors */
#define WFS_ERR_SIU_INVALID_PORT (-(SIU_SERVICE_OFFSET + 1))
#define WFS_ERR_SIU_SYNTAX (-(SIU_SERVICE_OFFSET + 2))
#define WFS_ERR_SIU_PORT_ERROR (-(SIU_SERVICE_OFFSET + 3))

/*=====*/
/* SIU Info Command Structures and variables */
/*=====*/

typedef struct _wfs_siu_status
{
    WORD fwDevice;
    WORD fwSensors [WFS_SIU_SENSORS_SIZE];
    WORD fwDoors [WFS_SIU_DOORS_SIZE];
    WORD fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR lpszExtra;
} WFSSIUSTATUS, * LPWFSSIUSTATUS;

typedef struct _wfs_siu_caps
```

```

{
    WORD            wClass;
    WORD            fwType;
    WORD            fwSensors [WFS_SIU_SENSORS_SIZE];
    WORD            fwDoors [WFS_SIU_DOORS_SIZE];
    WORD            fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD            fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD            fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR           lpszExtra;
} WFS_SIU_CAPS, * LPWFS_SIU_CAPS;

/*=====*/
/* SIU Execute Command Structures */
/*=====*/

typedef struct _wfs_siu_enable
{
    WORD            fwSensors [WFS_SIU_SENSORS_SIZE];
    WORD            fwDoors [WFS_SIU_DOORS_SIZE];
    WORD            fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD            fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD            fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR           lpszExtra;
} WFS_SIU_ENABLE, * LPWFS_SIU_ENABLE;

typedef struct _wfs_siu_set_ports
{
    WORD            fwDoors [WFS_SIU_DOORS_SIZE];
    WORD            fwIndicators [WFS_SIU_INDICATORS_SIZE];
    WORD            fwAuxiliaries [WFS_SIU_AUXILIARIES_SIZE];
    WORD            fwGuidLights [WFS_SIU_GUIDLIGHTS_SIZE];
    LPSTR           lpszExtra;
} WFS_SIU_SET_PORTS, * LPWFS_SIU_SET_PORTS;

typedef struct _wfs_siu_set_door
{
    WORD            wDoor;
    WORD            fwCommand;
} WFS_SIU_SET_DOOR, * LPWFS_SIU_SET_DOOR;

typedef struct _wfs_siu_set_indicator
{
    WORD            wIndicator;
    WORD            fwCommand;
} WFS_SIU_SET_INDICATOR, * LPWFS_SIU_SET_INDICATOR;

typedef struct _wfs_siu_set_auxiliary
{
    WORD            wAuxiliary;
    WORD            fwCommand;
} WFS_SIU_SET_AUXILIARY, * LPWFS_SIU_SET_AUXILIARY;

typedef struct _wfs_siu_set_guidlight
{
    WORD            wGuidLight;
    WORD            fwCommand;
} WFS_SIU_SET_GUIDLIGHT, * LPWFS_SIU_SET_GUIDLIGHT;

/*=====*/
/* SIU Message Structures */
/*=====*/

typedef struct _wfs_siu_port_event
{
    WORD            wPortType;
    WORD            wPortIndex;
    WORD            wPortStatus;
    LPSTR           lpszExtra;
}

```

```
} WFSSIUPORTEVENT, * LPWFSSIUPORTEVENT;

typedef struct _wfs_siu_port_error
{
    WORD            wPortType;
    WORD            wPortIndex;
    HRESULT         PortError;
    WORD            wPortStatus;
    LPSTR           lpszExtra;
} WFSSIUPORTERROR, * LPWFSSIUPORTERROR;

/* restore alignment */
#pragma pack (pop)

#ifdef __cplusplus
} /*extern "C"*/
#endif

#endif /* __INC_XFSSIU__H */
```