SOFTWARE INTEGRATION GUIDE

XFS PTR Service Provider

CUSTOM S.p.A.
Via Berettine 2/B
43010 Fontevivo (PARMA) - Italy
Tel. : +39 0521-680111
Fax : +39 0521-610701
http: www.custom.biz

Customer Service Department:
www.custom4u.it

XFS PTR LIBRARY

TABLE OF CONTENTS

INTRODUCTION

FORMS AND MEDIA

INFO COMMANDS

EXECUTE COMMANDS

EVENTS

INTRODUCTION

This service provider is supplied as a single 32 bit Windows DLL library (Custom_VKP_XFS.dll).

The Service Provider aims to be an XFS Service Provider in compliance with Extension for Financial Services (XFS) interface specification, Printer and Scanning Device Class (PTR), Type Receipt printer.

The minimum supported version is 3.0, the maximum supported version is 3.20.

This manual refers to version 3.20, prior versions can be still supported thanks to XFS backward compatibility (prior versions use a subset of the existing parameters).

The service provider requires the Windows print driver with "Status Monitor" or "Full" option selected of the affected printer to be installed on the target machine.

The Service Provider is fully compliant with the standard. This document describes some details about how standard compliance has been achieved and how options have been treated.

Models

NOMENCLATURE	DESCRIPTION	FIRMWARE RELEASE
VKP80II	Model with USB and RS232 ports.	3.29 or later
VKP80II SX	Model with USB and RS232 ports.	4.48 or later
VKP80III	Models with lateral, rear connectors with USB and RS232 port. Only in VKP80II emulation.	6.04 or later
KPM216HII-ETH	Model KPM216HII-ETH with horizontal paper mouth and 200 dpi printhead.	1.63 or later
	Model KPM216HII-ETH with horizontal paper mouth and 300 dpi printhead, Ethernet port.	1.69 or later
KPM216HIII	Model KPM216HIII with horizontal paper mouth and 200 dpi printhead. Model KPM216HIII with horizontal paper mouth and 300 dpi printhead.	1.96 or later

Driver

Custom VKP80II	FULL Driver (Package Rel. 2.18)
CUSTOM KPM216HII-ETH (200dpi)	STATUS MONITOR driver (Package Rel. 5.01)
CUSTOM KPM216HII-ETH (300dpi)"	STATUS MONITOR driver (Package Rel. 5.01)

Operating system

Windows XP (32bits)
Windows Seven (32 / 64 bits)
Windows 10 (32 / 64 bits)

XFS manager 3.2

Microsoft — Version 3.0.0.0

Known issues

Printouts with VKP80II, VKP80II SX and VKP80III might be longer than the length defined into the file.

The minimum physical receipt length depends on the hardware capabilities of the printers:
VKP80 models: 60mm
KPM216 models: 150mm

KPM216 and VKP80 models print drivers cannot dynamically set an arbitrary paper size. We can change the paper size changing the printer option from the operating system menu only: in this case the selected size will be used and the media file will be ignored. It is possible to use a “Roll” paper size too: with this option, the printer will manage independently the paper size according to the printing size.

After installing the Windows drivers, the service provider needs to be installed in a proper folder in the host PC, moreover XFS foresees that configuration is stored in the Windows registry together with the information required to locate the service provider.

Make sure that "Notch / B.Mark" parameter is set to "Disabled" in Printer Setup.

The printer has no capability to print multiple forms on the same sheet.

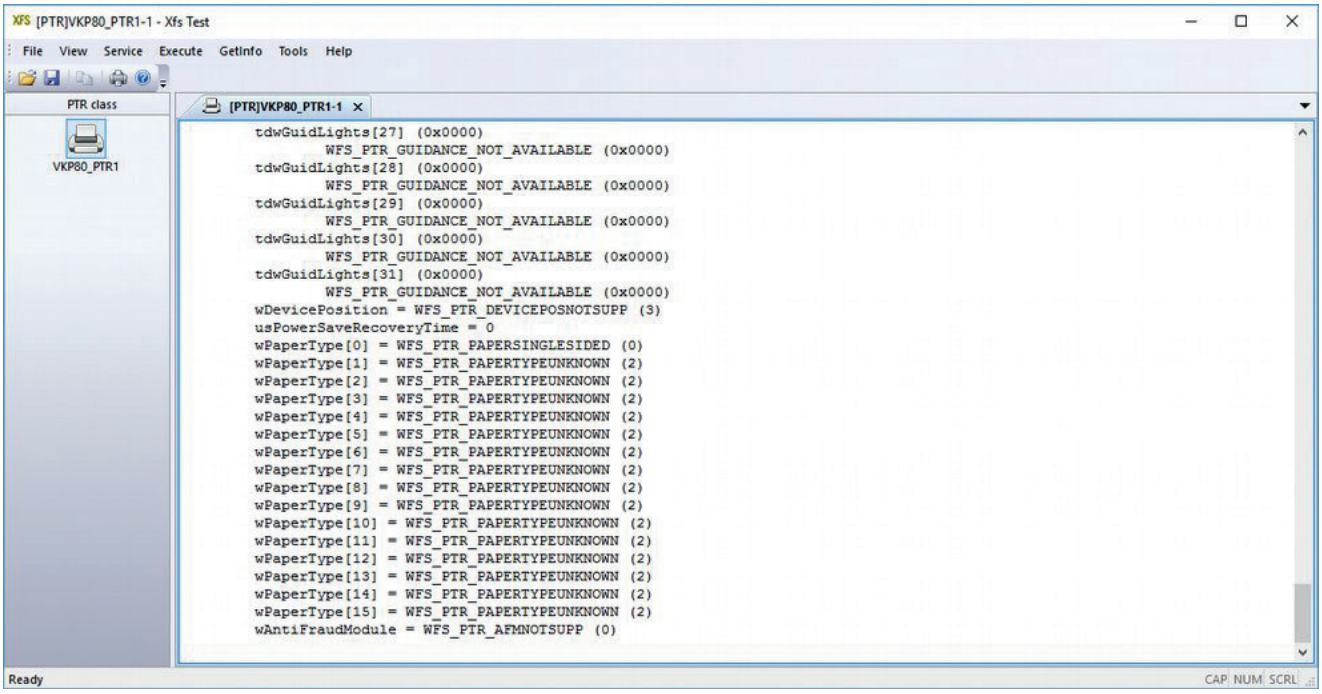
If Form defines a proportional font (such as “Arial”) the POINTSIZE parameter need to be specified, if a non proportional font is used (such as “Courier”), POINTSIZE or CPI and LPI need to be defined. A default non proportional font can be specified in the Registry.

Quick start

Before installing the service provider you need to install the printer drivers and an XFS manager (such as SDK320.msi from Microsoft).

The service provider comes with an installer and a test tool. By installing XFsTest Setup.msi you will have installed the service provider and a test tool.
If you have installed a very old previous version you might need to manually delete the registry branches:
HKEY_CURRENT_USER\XFS\LOGICAL_SERVICES\VKP80_PTR1
HKEY_CURRENT_USER\XFS\PHYSICAL_SERVICES\VKP80_PTR1

After proper installation you will get the following tool: Custom/XFS Test in the Program menu.
Click the printer icon and execute XFS functions (start with Service Open).
Tools>PrintTestForm is a one click complete test.
You will see XFS commands, data and events in the log window.



```
        tdwGuidLights[27] (0x0000)
                WFS_PTR_GUIDANCE_NOT_AVAILABLE (0x0000)
        tdwGuidLights[28] (0x0000)
                WFS_PTR_GUIDANCE_NOT_AVAILABLE (0x0000)
        tdwGuidLights[29] (0x0000)
                WFS_PTR_GUIDANCE_NOT_AVAILABLE (0x0000)
        tdwGuidLights[30] (0x0000)
                WFS_PTR_GUIDANCE_NOT_AVAILABLE (0x0000)
        tdwGuidLights[31] (0x0000)
                WFS_PTR_GUIDANCE_NOT_AVAILABLE (0x0000)
        wDevicePosition = WFS_PTR_DEVICEPOSNOTSUPP (3)
        usPowerSaveRecoveryTime = 0
        wPaperType[0] = WFS_PTR_PAPERSINGLESIDED (0)
        wPaperType[1] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[2] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[3] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[4] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[5] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[6] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[7] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[8] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[9] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[10] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[11] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[12] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[13] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[14] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wPaperType[15] = WFS_PTR_PAPERTYPEUNKNOWN (2)
        wAntiFraudModule = WFS_PTR_AFMNOTSUPP (0)
```

Manual Configuration

The user dependent configuration information are loaded by the Service Provider when a new service is opened. The configurations are stored in the Windows Registry.
Most folder paths contain the environment variable “%APPDATA%” since its value is depending from the Windows version and local. This special folder is where applications store their data and it can be expanded typing: “%APPDATA%” in the address bar of Windows Explorer.
The paths stored in the Windows Registry must be separated by a double backslash according to the C language escape notation.
Each printer is known by a logical name used when opening a service and each logical name owns a configuration.

An example of a file with the configuration for a Custom VKP80II with the logical name “VKP80_PTR1”:

```
Windows Registry Editor Version 5.00
; Perform a search and replace for the following information to be personalised
; "name" -> "the windows name of the printer"
; "c:\\Custom\\Custom_VKP_XFS.dll" -> "real path to DLL"
; "c:\\Custom\\forms" -> "real path to form folder
; "c:\\Custom\\medias" -> "real path to nedia folder

; Default user subkeys

[HKEY_USERS\.DEFAULT\XFS\LOGICAL_SERVICES\VKP80_PTR1]
"class"="PTR"
"provider"="VKP80_PTR"

[HKEY_USERS\.DEFAULT\XFS\PHYSICAL_SERVICES\VKP80_PTR1]
"name" = "CUSTOM VKP80 II"
"device" = "VKP80II"

; Current user subkey (same as default user)
[HKEY_CURRENT_USER\XFS\LOGICAL_SERVICES\VKP80_PTR1]
"class"="PTR"
"provider"="VKP80_PTR"

[HKEY_LOCAL_MACHINE\SOFTWARE\XFS\PHYSICAL_SERVICES\VKP80_PTR1]
"name" = "CUSTOM VKP80 II"
"maxretract1" = "15"
"device" = "VKP80II"
"formpath"="c:\\Custom\\forms"
"formext"="flc"
"mediapath"="c:\\Custom\\media"
"mediaext"="mlc"
"settings"="1"

;if settings="1" WFS_PTR_MEDIATAKEN is always sent if ticket out sensor status changes
;if settings="0" WFS_PTR_MEDIATAKEN is sent only if a form has been printed

; Local machine subkey
[HKEY_LOCAL_MACHINE\SOFTWARE\XFS\SERVICE_PROVIDERS\VKP80_PTR]
"dllname"="c:\\Custom\\Custom_VKP_XFS.dll"
"vendorname"="Custom"
"version"="1.11"

;RAW_DATA header and footer, enable if needed and modify data with this format "HEX HEX HEX"
;[HKEY_LOCAL_MACHINE\SOFTWARE\XFS\PHYSICAL_SERVICES\VKP80_PTR1]
;"rawdataheader"="0A"
;"rawdatatrailer"="0A"
```

The XFS Manager uses its configuration information to define the relationships among the applications and the Service Providers. Service Providers need to configure a pointer to the service provider DLL in the HKEY_LOCAL_MACHINE registry branch. The “dllname” key shall match the real folder and file name (complete path) where the DLL is installed as defined by the XFS standard.
The PHYSICAL_SERVICES keys contain several proprietary settings to configure and customize the SP.
Some keys are optional, if omitted the default values will be used. These keys are described below.

Name

The name of the Windows printer driver. This field is mandatory.
This value shall match the Windows driver name and must be selected according to the printer model (“device” key).

Device

The device name for the Service Provider, must be set according to the printer model and the “name” key.
This field is mandatory.

“name” key	“device” key
CUSTOM VKP80II	VKP80II
CUSTOM KPM216HII-ETH (300 dpi)	KPM216HUP300
CUSTOM KPM216HII-ETH (200 dpi)	KPM216HUP200

Maxretract1

Set the threshold for the retract bin full alarm. The default is 0 (no alarm).

Highretract1

Set the threshold for the retract nearly full alarm. The default is 0 (no alarm).

Formpath

The path where the form files are searched and stored. The default path is “%APPDATA%\\Custom\\PTR_XFS”.

Formext

The file extension used for form files. The default extension is “flc”.

Mediapath

The path where the media files are searched and stored. The default path is “%APPDATA%\\Custom\\PTR_XFS”.

Mediaext

The file extension used for media files. The default extension is “mlc”.

CPI

The Character per inches printer density emulated by the SP when the UNITBASE is set to ROWCOLUMN.
Default value is 13.5 CPI. This parameter allows to use MEDIA defined in ROWCOLUMN adapting the printer density to the width of the paper.

LPI

The Character per inches printer density emulated by the SP when the UNITBASE is set to ROWCOLUMN.
Default value is 8.0 LPI. This parameter allows to use MEDIA defined in ROWCOLUMN adapting the printer density to the height of the paper.

Font

This is the default font if a XFSFIELD does not define the FONT field. This key value must be the name of an installed Windows font, furthermore if the UNITBASE is set to ROWCOLUMN the font must be mono-spaced (non-proportional). The default value is “Arial”.

Flags

A 32 bit value in hexadecimal notation used to activate some options. The default is “00000000”. This is currently reserved and normally not to be used; it will only be manipulated in case of problems guaranteeing compatibility with multiple applications.

Rawdataheader

The string associated to this key defines some binary data to be sent before the data specified in the command WFS_CMD_PTR_RAW_DATA. These data are sent to the printer only if the value of the command parameter wInputData is WFS_PTR_NOINPUTDATA.
The string must be formatted as string of hexadecimal bytes separated by spaces (I.E. “30 31 32 0D 0A”). The default value is an empty string.

Rawdatatrailer

The string associated to this key defines some binary data to be sent after the data specified in the command WFS_CMD_PTR_RAW_DATA. These data are sent to the printer only if the value of the command parameter wInputData is WFS_PTR_NOINPUTDATA.
The string must be formatted as string of hexadecimal bytes separated by spaces (I.E. “30 31 32 0D 0A”). The default value is an empty string.

Logging

All commands and events are traced for diagnostic purpose in a log file. The log file full path is:“%APPDATA%\\Custom\\PTR_XFS \\PTR_XFS.log”.
This file must be supplied when reporting software problem.

FORMS AND MEDIA

XfsForms

All tags that are normally used in ATMs and are supported by printer hardware are implemented according to the standard. There are some tags, defined as optional by the standard, that are not applicable either because their behaviour cannot be obtained with the printer hardware or because they have very limited application and not wide acceptance and common community understanding.
The development has been made trying to achieve the widest possible compatibility with existing forms and it has been tested against all available forms from real life ATM applications.
Since the XfsForms language is very wide and covers all different scenarios and hardware options it is the area where most “interpretation” of the standard is necessary.
In particular:

- XfsFrame, which is optional for the standard and not widely used, is not defined and any XfsFrame tags are ignored, if there are fields associated to the frame they are not assigned.
- Other tags not implemented and ignored are: SKEW, FOLLOWS, BARCODE, COERCIVITY, ACCESS.
- Most parameters of the tag STYLE are ignored too, the managed parameters are: NORMAL, BOLD, ITALIC, UNDER, STRIKETHROUGH.
- Some parameters of the OVERLOW tag are managed partially. TERMINATE and TRUNCATE both print the field up to the last character that is inside the layout rectangle; BESTFIT prints the field up to the last word that is inside the layout rectangle, then insert an ellipsis “(...)”; the tag OVERWRITE and WORDWRAP are ignored.

When printing text fields, it is possible to print using non proportional fonts, in addition to setting the POINTSIZE to 0, if the font name refers to a mono-spaced font present in the system, the tag LPI and CPI will define the attributes of the font. If the font cannot be selected for printing, a default font with point-size 4 will be used and no warning event will be generated.
If printing does not fit in the layout, it is cropped as foreseen by the standard.
Be, therefore, careful choosing the POINTSIZE value (or the CPI and LPI values when POINTSIZE value is 0). The printing is possible only inside the layout rectangle and a part or whole the text could not be printed. The layout rectangle is indicated by the tag SIZE.
It is possible to define more than one XfsForm in a single file.

XfsMedia

XFSMedia are used to set the paper size to be used for printing. The only used tags are XFSMEDIA, BEGIN, END, UNIT, SIZE, all others tags will be ignored. The unit ROWCOLUMN is equivalent to 13.3 CPI and 8 LPI.
It is possible to define more than one XfsMedia in a file.

- Example of media (80 mm of width and 100 mm of height).

```
XFSMEDIA "TICKET 80x100"

BEGIN
    UNIT MM, 1, 1
    SIZE 80, 100
END
```

- Example of media (3 inches of width and 4 inches and 5/10 of height).

```
XFSMEDIA "TICKET 3"x4""

BEGIN
    UNIT INCH, 10, 10
    SIZE 30, 45
END
```

- Example of media (Standard A4 size).

```
XFSMEDIA "A4 sheet 210x297mm"

BEGIN
    UNIT MM, 1, 1
    SIZE 210, 297
END
```

- Example of media (Standard A5 size).

```
XFSMEDIA "A5 sheet 210x148mm"

BEGIN
    UNIT MM, 1, 1
    SIZE 210, 148
END
```

INFO COMMANDS

WFS_INF_PTR_STATUS

[Input parameters]	None.

[Output parameters]

```
LPWFSPTRSTATUS lpStatus
typedef struct _wfs_ptr_status {
        WORD fwDevice;
        WORD fwMedia;
        WORD fwPaper[WFS_PTR_SUPPLYSIZE];
        WORD fwToner;
        WORD fwInk;
        WORD fwLamp;
        LPWFSPTRRETRACTBINS *lppRetractBins;
        USHORT usMediaOnStacker;
        LPSTR lpszExtra;
        DWORD dwGuidLights[WFS_PTR_GUIDLIGHTS_SIZE];
        WORD wDevicePosition;
        USHORT usPowerSaveRecoveryTime;
        WORD wPaperType[WFS_PTR_SUPPLYSIZE];
        WORD wAntiFraudModule;
} WFSPTRSTATUS, *LPWFSPTRSTATUS;
```

Where:

PARAMETER	MEANING	
fwDevice	WFS_PTR_DEVONLINE	The device is online.
	WFS_PTR_DEVOFFLINE	The device is powered off or not connected.
	WFS_PTR_DEVHWERROR	The device is inoperable.
	WFS_PTR_DEVUSERERROR	The device is inoperable because the printer cover is open.
fwMedia	WFS_PTR_MEDIAPRESENT	The ticket is engaging the presenter sensor.
	WFS_PTR_MEDIANOTPRESENT	The ticket is not engaging the presenter sensor.
	WFS_PTR_MEDIAUNKNOWN	The presenter sensor status cannot be determined.

fwPaper[...]	All paper supply statuses are set to WFS_PTR_PAPERNOTSUPP but WFS_PTR_SUPPLYUPPER: fwPaper[WFS_PTR_SUPPLYUPPER]	
	WFS_PTR_PAPEROUT	The paper supply is empty.
	WFS_PTR_PAPERLOW	The paper supply is low.
	WFS_PTR_PAPERFULL	The paper supply is full.
	WFS_PTR_PAPERNOTSUPP	The capability to report the state is not supported because the paper supply does not exist.
fwToner	WFS_PTR_TONERNOTSUPP	The capability is not supported.
fwInk	WFS_PTR_INKNOTSUPP	The capability is not supported.
fwLamp	WFS_PTR_LAMPNOTSUPP	The capability is not supported.
lppRetractBins	Pointer to an array of two pointers to WFSPTRRETRACTBINS structure. The first one points to the structure related to the retract bin, the last one is a NULL pointer.	

```
typedef struct _wfs_ptr_retract_bins {
            WORD wRetractBin;
            USHORT usRetractCount;
} WFSPTRRETRACTBINS, *LPWFSPTRRETRACTBINS;
```

wRetractBin	WFS_PTR_RETRACTBINOK	The retract bin is empty.
	WFS_PTR_RETRACTBINHIGH	The retract bin has reached its nearly full threshold.
	WFS_PTR_RETRACTBINFULL	The retract bin has reached its full threshold.
usRetractCount	The number of media retracted to this bin. This value is persistent; it may be reset to zero by the WFS_CMD_PTR_RESET_COUNT command.	
usMediaOnStacker	Not applicable, always set to zero.	
lpszExtra	No extra data are defined, always returning a NULL pointer.	
dwGuidLights[...]	Guidance lights are managed by the printer, all guidance light indicators always set to WFS_PTR_GUIDANCE_NOT_AVAILABLE.	
wdevicePosition	The printer has not the capability to of detecting its position, always set to WFS_PTR_DEVICEPOSNOTSUPP.	
usPowerSaveRecoveryTime	Power save control is not supported, always set to zero.	
wpaperType[...]	All paper types are set to WFS_PTR_PAPERNOTSUPP but WFS_PTR_SUPPLYUPPER. wpaperType[WFS_PTR_SUPPLYUPPER] always set to WFS_PTR_PAPERSINGLESIDED.	
wantiFraudModule	No anti-fraud module is available, always set to WFS_PTR_AFMNOTSUPP.	

[Specific error codes] None.

[Specific events] None.

WFS_INF_PTR_CAPABILITIES

[Input parameters] None.

[Output parameters]

```
LPWFSPTRSTATUS lpStatus;
typedef struct _wfs_ptr_caps {
            WORD wClass;
            WORD fwType;
            BOOL bCompound;
            WORD wResolution;
            WORD fwReadForm;
            WORD fwWriteForm;
            WORD fwExtents;
            WORD fwControl;
            USHORT usMaxMediaOnStacker;
            BOOL bAcceptMedia;
            BOOL bMultiPage;
            WORD fwPaperSources;
            BOOL bMediaTaken;
            USHORT usRetractBins;
            LPUSHORT lpusMaxRetract;
            WORD fwImageType;
            WORD fwFrontImageColorFormat;
            WORD fwBackImageColorFormat;
            WORD fwCodelineFormat;
            WORD fwImageSource;
            WORD fwCharSupport;
            BOOL bDispensePaper;
            LPSTR lpszExtra;
            DWORD dwGuidLights[WFS_PTR_GUIDLIGHTS_SIZE];
            LPSTR lpszWindowsPrinter;
            BOOL bMediaPresented;
            USHORT usAutoRetractPeriod;
            BOOL bRetractToTransport;
            BOOL bPowerSaveControl;
            WORD fwCoercivityType;
            WORD fwControlPassbook;
            WORD wPrintSides;
            BOOL bAntiFraudModule;
} WFSPTRCAPS, *LPWFSPTRCAPS;
```

Where:

PARAMETER	MEANING
wClass	Specifies the logical service class as WFS_SERVICE_CLASS_PTR.
fwType	Specifies the type of device as WFS_PTR_TYPERECEIPT.
bCompound	The printer is not part of a compound device, always set to FALSE.
wResolution	Always returning WFS_PTR_RESHIGH.
fwReadForm	The printer has not read media capabilities, always set to zero.
fwWriteForm	The printer can print text and graphics, always set to the arithmetic OR of the flags WFS_PTR_WRITETEXT and WFS_PTR_WRITEGRAPHICS.
fwExtents	The printer does not accept media, always set to zero.

fwControl	The printer can eject, expel or retract ticket and wait for data flush. The allowed flags are WFS_PTR_CTRLEJECT, WFS_PTR_CTRLRETRACT, WFS_PTR_CTRLEXPEL and WFS_PTR_CTRLFLUSH.
usMaxMediaOnStacker	Not applicable, always set to zero.
bAcceptMedia	Not applicable, always set to FALSE.
bMultiPage	Printer can not handle multiple page print jobs, always set to FALSE. In order to keep the interface coherent, each WFS_CMD_PTR_PRINT_FORM should fit in a page: a multipage receipt will be obtained by printing multiple forms.
fwPaperSources	Printer has a single paper source, always set to WFS_PTR_PAPERUPPER.
bMediaTaken	The printer can detect when the ticket is taken by the user and raise a WFS_SRVE_MEDIATAKEN event, always set to TRUE.
usRetractBins	Number of retract bins, always set to 1.
lpusMaxRetract	Points to an unsigned short value defining the maximum number of tickets that the retract bin can hold. This value is loaded from the Service Provider configuration.
fwImageType	Printer does not support imaging, always set to zero.
fwFrontImageColorFormat	Printer does not support imaging, always set to zero.
fwBackImageColorFormat	Printer does not support imaging, always set to zero.
fwCodelineFormat	Printer does not support codeline reading, always set to zero.
fwImageSource	Printer does not support imaging, always set to zero.
fwCharSupport	The printer can manage ASCII and UNICODE forms, always set to the arithmetic OR of the flags WFS_PTR_ASCII and WFS_PTR_UNICODE.
bDispensePaper	Printer cannot dispense paper, always set to FALSE.
lpszExtra	No extra data are defined, always returning a NULL pointer.
dwGuidLights[...]	Guidance lights are managed by the printer, all guidance light indicators always set to WFS_PTR_GUIDANCE_NOT_AVAILABLE.
lpszWindowsPrinter	Return the Windows name of the printer. I.E. “CUSTOM VKP80 II”.
lpszExtra	No extra data are defined, always returning a NULL pointer.
bMediaPresented	Always set to a FALSE: media presented is not notified, media is presented automatically after printing.
usAutoRetractPeriod	The printer is not able to retract ticket after a time-out, always set to zero.
bRetractToTransport	The printer is not able to retract an ejected ticket to the transport, always set to FALSE.
bPowerSaveControl	Power save control is not supported, always set to FALSE.
fwCoercivityType	The printer cannot write magnetic stripes, always set to WFS_PTR_COERCIVITYNOTSUPP.
fwControlPassbook	The printer cannot handle passbooks, always set to PBKCTRLNOTSUPP.
wPrintSides	Not applicable, always set to WFS_PTR_PRINTSIDESNOTSUPP.
bAntiFraudModule	No anti-fraud module is available, always set to FALSE.

[Specific error codes]	None.
[Specific events]	None

WFS_INF_PTR_FORM_LIST

[Input parameters] None.

[Output parameters] *LPSTR lpszFormList*

Where:

PARAMETER	MEANING
lpszFormList	Pointer to a list of null-terminated form names, with the final name terminating with two null characters.

[Specific error codes] Always returns WFS_SUCCESS or a generic error code.

[Specific events] None.

WFS_INF_PTR_MEDIA_LIST

[Input parameters]	None.
[Output parameters]	*LPSTR lpszMediaList*

Where:

PARAMETER	MEANING
lpszMediaList	Pointer to a list of null-terminated media names, with the final name terminating with two null characters.

[Specific error codes]	Always returns WFS_SUCCESS
[Specific events]	None.

WFS_INF_PTR_QUERY_FORM

[Input parameters] *LPSTR lpszFormName*

Where:

PARAMETER	MEANING
lpszFormName	Points to the null-terminated form name on which to retrieve details.

[Output parameters]

```
LPWFSFRMHEADER lpHeader
typedef struct _wfs_frm_header {
        LPSTR lpszFormName;
        WORD wBase;
        WORD wUnitX;
        WORD wUnitY;
        WORD wWidth;
        WORD wHeight;
        WORD wAlignment;
        WORD wOrientation;
        WORD wOffsetX;
        WORD wOffsetY;
        WORD wVersionMajor;
        WORD wVersionMinor;
        LPSTR lpszUserPrompt;
        WORD fwCharSupport;
        LPSTR lpszFields;
        WORD wLanguageID;
} WFSFRMHEADER, *LPWFSFRMHEADER;
```

[Specific error codes]

WFS_ERR_PTR_FORMNOTFOUND	The specified form cannot be found.
WFS_ERR_PTR_FORMINVALID	The specified form is invalid.

[Specific events] None.

WFS_INF_PTR_QUERY_MEDIA

[Input parameters] *LPSTR lpszMediaName;*

Where:

PARAMETER	MEANING
lpszMediaName	Pointer to the null-terminated media name about which to retrieve details.

[Output parameters] *LPWFSFRMMEDIA lpMedia*

```
typedef struct _wfs_frm_media {
        WORD fwMediaType;
        WORD wBase;
        WORD wUnitX;
        WORD wUnitY;
        WORD wSizeWidth;
        WORD wSizeHeight;
        WORD wPageCount;
        WORD wLineCount;
        WORD wPrintAreaX;
        WORD wPrintAreaY;
        WORD wPrintAreaWidth;
        WORD wPrintAreaHeight;
        WORD wRestrictedAreaX;
        WORD wRestrictedAreaY;
        WORD wRestrictedAreaWidth;
        WORD wRestrictedAreaHeight;
        WORD wStagger;
        WORD wFoldType;
        WORD wPaperSources;
} WFSFRMMEDIA, *LPWFSFRMMEDIA;
```

[Specific error codes]

WFS_ERR_PTR_MEDIANOTFOUND	The specified medium cannot be found.
WFS_ERR_PTR_MEDIAINVALID	The specified medium is invalid.

[Specific events] None.

WFS_INF_PTR_QUERY_FIELD

[Input parameters]

```
LPWFSPTRQUERYFIELD lpQueryField
typedef struct _wfs_ptr_query_field {
          LPSTR lpszFormName;
          LPSTR lpszFieldName;
} WFSPTRQUERYFIELD, *LPWFSPTRQUERYFIELD;
```

[Output parameters]

```
LPWFSFRMFIELD *lppFields;
```

Where:

PARAMETER	MEANING
lppFields	Pointer to a null-terminated array of pointers to WFSFRMFIELD structures:

```
typedef struct _wfs_frm_field {
LPSTR lpszFieldName;
            WORD wIndexCount;
            WORD fwType;
            WORD fwClass;
            WORD fwAccess;
            WORD fwOverflow;
            LPSTR lpszInitialValue;
            LPWSTR lpszUNICODEInitialValue;
            LPSTR lpszFormat;
            LPWSTR lpszUNICODEFormat;
            WORD wLanguageID;
            WORD wCoercivity;
} WFSFRMFIELD, *LPWFSFRMFIELD;
```

[Specific error codes]

WFS_ERR_PTR_FORMNOTFOUND	The specified form cannot be found.
WFS_ERR_PTR_FIELDNOTFOUND	The specified field cannot be found.
WFS_ERR_PTR_FORMINVALID	The specified form is invalid.

[Specific events] None.

WFS_INF_PTR_CODELINE_MAPPING

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_CONTROL_MEDIA

[Input parameters] *LPDWORD lpdwMediaControl*

Where:

PARAMETER	MEANING
lpdwMediaControl	Pointer to a value which specifies the manner in which the media should be handled, as an arithmetic OR of the following flags. An immediate error WFS_ERR_INVALID_DATA is given in return if we combine the flags WFS_PTR_CTRLEJECT, WFS_PTR_CTREXPEL and WFS_PTR_CTRLRETRACT with each other.
	WFS_PTR_CTRLFLUSH: Flush any data to the printer.
	WFS_PTR_CTRLEJECT: Flush any data to the printer, then throw the media out instead of expelling it (a behaviour different from standard has been implemented as customer's requirement).
	WFS_PTR_CTREXPEL: Flush any data to the printer, then throw the media out.
	WFS_PTR_CTRLRETRACT: Flush any data to the printer, then retract the media.

[Output parameters] None.

[Specific error codes]

WFS_ERR_PTR_NOMEDIAPRESENT	The control action could not be completed because there is no media in the device, the media is not in a position where it can be controlled, or (in the case of WFS_PTR_CTRLRETRACT) has been removed.
WFS_ERR_PTR_RETRACTBINFULL	The retract bin is full. No more media can be retracted. The current media is still in the printer presenter.
WFS_ERR_PTR_PAPERJAMMED	The paper is jammed.

[Specific events]

WFS_USRE_PTR_RETRACTBINTHRESHOLD	The retract bin is full; operator intervention is required. Note that this event is sent only once, at the point at which the bin becomes full or nearly full. It is sent with WFS_PTR_RETRACTBINFULL or WFS_PTR_RETRACTBINHIGH status.
WFS_SRVE_PTR_MEDIATAKEN	The media has been taken by the user.

WFS_USRE_PTR_PAPERTHRESHOLD	The paper supply is low or empty; operator intervention is required. Note that this event is sent only once, at the point at which the supply becomes low or empty. It is sent with WFS_PTR_PAPERLOW or WFS_PTR_PAPEROUT status.

WFS_CMD_PTR_PRINT_FORM

[Input parameters]

```
LPWFSPTRPRINTFORM lpPrintForm
ypedef struct _wfs_ptr_print_form {
        LPSTR lpszFormName;
        LPSTR lpszMediaName;
        WORD wAlignment;
        WORD wOffsetX;
        WORD wOffsetY;
        WORD wResolution;
        DWORD dwMediaControl;
        LPSTR lpszFields;
        LPWSTR lpszUNICODEFields;
        WORD wPaperSource;
} WFSPTRPRINTFORM, *LPWFSPTRPRINTFORM;
```

Where:

PARAMETER	MEANING	
lpszFormName	Pointer to the null-terminated form name.	
lpszMediaName	Pointer to the null-terminated media name. It can be a NULL pointer: in this case the form size will be calculated by the printer driver according to the printed area size.	
wAlignment	WFS_PTR_ALNUSEFORMDEFN	Use the alignment specified in the form definition.
	WFS_PTR_ALNTOPLEFT	Align form to top left of physical medium.
	WFS_PTR_ALNTOPRIGHT	Align form to top right of physical medium.
	WFS_PTR_ALNBOTTOMLEFT	Align form to bottom left of physical medium.
	WFS_PTR_ALNBOTTOMRIGHT	Align form to bottom right of physical medium.
wOffsetX	Specifies the horizontal offset of the form. A value of WFS_PTR_OFFSETUSEFORMDEFN indicates that the x-offset value from the form definition should be used.	
wOffsetY	Specifies the vertical offset of the form. A value of WFS_PTR_OFFSETUSEFORMDEFN indicates that the y-offset value from the form definition should be used.	
wResolution	This parameter is ignored by the Service Provider.	
dwMediaControl	Pointer to a value which specifies the manner in which the media should be handled, as an arithmetic OR of the following flags. An immediate error WFS_ERR_INVALID_DATA is given in return if we combine the flags WFS_PTR_CTRLEJECT, WFS_PTR_CTREXPEL and WFS_PTR_CTRLRETRACT with each other.	
	WFS_PTR_CTRLFLUSH	Flush any data to the printer.
	WFS_PTR_CTRLEJECT	Flush any data to the printer, then throw the media out instead of expelling it (a behaviour different from standard has been implemented as customer's requirement).
	WFS_PTR_CTREXPEL	Flush any data to the printer, then throw the media out.
	WFS_PTR_CTRLRETRACT	Flush any data to the printer, then retract the media.

lpszFields	Pointer to a series of "<FieldName>=<FieldValue>" ASCII encoded strings, where each string is null-terminated with the entire field string terminating with two null characters. The Service provider, regardless the declared form encoding (fwCharSupport field in WFSFRMHEADER structure), tries to load the fields from the lpszFields pointer as ASCII strings, if the pointer is NULL, then tries from the lpszUNICODEFields pointer as UNICODE.
lpszUNICODEFields	Pointer to a series of "<FieldName>=<FieldValue>" UNICODE encoded strings, where each string is nullterminated with the entire field string terminating with two null characters. The Service provider, regardless the declared form encoding (fwCharSupport field in WFSFRMHEADER structure), tries to load the fields from the lpszFields pointer as ASCII strings, if the pointer is NULL, then tries from the lpszUNICODEFields pointer as UNICODE.
wPaperSource	Specifies the Paper source to use when printing this form: it must be WFS_PTR_PAPERANY or WFS_PTR_PAPERUPPER.

[Output parameters] None.

[Specific error codes]

WFS_ERR_PTR_FORMNOTFOUND	The specified form definition cannot be found.
WFS_ERR_PTR_MEDIAOVERFLOW	The form overflowed the media.
WFS_ERR_PTR_FIELDSPECFAILURE	The syntax of the lpszFields or lpszUNICODEFields member is invalid WFS_ERR_PTR_FIELDERROR An error occurred while processing a field, causing termination of the print request. An execute event WFS_EXEE_PTR_FIELDERROR is posted with the details
WFS_ERR_PTR_MEDIANOTFOUND	The specified media definition cannot be found.
WFS_ERR_PTR_MEDIAINVALID	The specified media definition is invalid.
WFS_ERR_PTR_FORMINVALID	The specified form definition is invalid.
WFS_ERR_PTR_SOURCEINVALID	The selected paper source is not supported by the hardware.
WFS_ERR_PTR_PAPEROUT	The paper supply is empty.

Same error codes of command WFS_CMD_PTR_CONTROL_MEDIA can be generated too. Note that a WFS_ERR_NOMEDIAPRESENT event can be generated too, if the dwMediaControl flag WFS_PTR_CTRLRETRACT is set (retract after print).

[Specific events]

WFS_EXEE_PTR_FIELDERROR	A fatal error occurred while processing a field.

Same [Specific events] of command WFS_CMD_PTR_CONTROL_MEDIA can be generated too.

WFS_CMD_PTR_READ_FORM

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_RAW_DATA

This command is used to send raw data (a byte string of device dependent data) to the printer.

[Input parameters]

```
typedef struct _wfs_ptr_raw_data {
        WORD wInputData;
        ULONG ulSize;
        LPBYTE lpbData;
} WFSPTRRAWDATA, *LPWFSPTRRAWDATA;
```

Where:

PARAMETER	MEANING
wInputData	Specifies if are expected data in response from the printer. WFS_PTR_NOINPUTDATA No input data is expected. WFS_PTR_INPUTDATA Input data is expected.
ulSize	Specifies the size in bytes of the string send to the printer.
lpbData	Pointer to the data to be sent to the printer.

[Output parameters]

```
typedef struct _wfs_ptr_raw_data_in {
        ULONG ulSize;
        LPBYTE lpbData;
} WFSPTRRAWDATAIN, *LPWFSPTRRAWDATAIN;
```

Where:

PARAMETER	MEANING
ulSize	Size in bytes of the string received from the printer. It is always zero if wInputData is WFS_PTR_NOINPUTDATA.
lpbData	Pointer to the data received from the printer.

[Specific error codes]

WFS_ERR_PTR_PAPEROUT	The paper supply is empty.

[Specific events]

None.

WFS_CMD_PTR_MEDIA_EXTENTS

This command is not implemented.

Iimmediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_RESET_COUNT

[Input parameters] *LPUSHORT lpusBinNumber;*

Where:

PARAMETER	MEANING
lpusBinNumber	To reset the number of retracted items, you can either make it point to a value of 1, or assign it to NULL.

[Output parameters] None.

[Specific error codes] It always returns WFS_SUCCESS.

[Specific events]

WFS_USRE_PTR_RETRACTBIN-THRESHOLD	The status of the retract bin has changed from high or full to a good state. The event is sent with WFS_PTR_RETRACTBINOK status.

WFS_CMD_PTR_READ_IMAGE

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_RESET

[Input parameters]

```
LPWFSPTRRESET lpReset;
typedef struct _wfs_ptr_reset {
        DWORD dwMediaControl;
        USHORT usRetractBinNumber;
} WFSPTRRESET, *LPWFSPTRRESET;
```

If lpReset is a NULL pointer, the reset command is sent immediately to the printer. A ticket possibly exposed in the presenter will be ejected.

Where:

<table>
<tr><th>PARAMETER</th><th colspan="2">MEANING</th></tr>
<tr><td rowspan="4">dwMediaControl</td><td colspan="2">To reset the number of retracted items, you can either make it point to a value of 1, or assign it to NULL.</td></tr>
<tr><td>WFS_PTR_CTRLEJECT</td><td>Throw out the media before performing the printer reset.</td></tr>
<tr><td>WFS_PTR_CTRLEXPEL</td><td>Throw out the media before performing the printer reset.</td></tr>
<tr><td>WFS_PTR_CTRLRETRACT</td><td>Retract the media before performing the printer reset.</td></tr>
<tr><td>usRetractBinNumber</td><td colspan="2">This value must be 1.</td></tr>
</table>

[Output parameters] None.

[Specific error codes]

WFS_ERR_PTR_RETRACTBINFULL	The retract bin is full. No more media can be retracted. The current media is still in the printer presenter.
WFS_ERR_PTR_PAPERJAMMED	The paper is jammed.

[Specific events]

WFS_SRVE_PTR_MEDIADETECTED	A media is detected in the device during a reset operation.
WFS_USRE_PTR_RETRACTBINTHRESHOLD	The retract bin is full or high; operator intervention is required. Note that this event is sent only once, at the oint at which the bin becomes full or high. It is sent with WFS_PTR_RETRACTBINFULL or WFS_PTR_RETRACTBINHIGH status.

WFS_CMD_PTR_RETRACT_MEDIA

Retract the media from the printer presenter to the internal bin.

[Input parameters] *LPUSHORT lpusBinNumber;*

Where:

PARAMETER	MEANING
lpusBinNumber	Pointer to the retract bin number, it must be 1.

[Output parameters] *LPUSHORT lpusBinNumber;*

Where:

PARAMETER	MEANING
lpusBinNumber	Pointer to the retract bin number where the media is deposited, it is always 1.

[Specific error codes]

WFS_ERR_PTR_NOMEDIAPRESENT	The control action could not be completed because there is no media in the device, the media is not in a position where it can be controlled, or has been removed.
WFS_ERR_PTR_RETRACTBINFULL	The retract bin is full. No more media can be retracted. The current media is still in the printer presenter.
WFS_ERR_PTR_PAPERJAMMED	The paper is jammed.

[Specific events]

WFS_USRE_PTR_RETRACTBINTHRESHOLD	The retract bin is full; operator intervention is required. Note that this event is sent only once, at the point at which the bin becomes full or nearly full. It is sent with WFS_PTR_RETRACTBINFULL or WFS_PTR_RETRACTBINHIGH status.
WFS_SRVE_PTR_MEDIATAKEN	The media has been taken by the user.
WFS_USRE_PTR_PAPERTHRESHOLD	The paper supply is low or empty; operator intervention is required. Note that this event is sent only once, at the point at which the supply becomes low or empty. It is sent with WFS_PTR_PAPERLOW or WFS_PTR_PAPEROUT status.

WFS_CMD_PTR_DISPENSE_PAPER

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_SET_GUIDANCE_LIGHT

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_PRINT_RAW_FILE

This command is used to print a file in the native printer language. This kind of file can be created on Windows checking the option “Print to file” on the printer dialogue box (Please note that some programs do not give you the option to print to file).

[Input parameters]

```
typedef struct _wfs_ptr_print_raw_file {
        LPSTR lpszFileName;
        DWORD dwMediaControl;
        DWORD dwPaperSource;
} WFSPTRPRINTRAWFILE, *LPWFSPTRPRINTRAWFILE;
```

Where:

PARAMETER	MEANING
lpszFileName	Pointer to the null-terminated file name Full path and file name of the file to be printer.
dwMediaControl	Value which specifies the manner in which the media should be handled, as an arithmetic OR of the following flags. An immediate error WFS_ERR_INVALID_DATA is given in return if we combine the flags WFS_PTR_CTRLEJECT, WFS_PTR_CTREXPEL and WFS_PTR_CTRLRETRACT with each other. WFS_PTR_CTRLFLUSH: Flush any data to the printer. WFS_PTR_CTRLEJECT: Flush any data to the printer, then throw the media out instead of expelling it (a behaviour different from standard has been implemented as customer's requirement). WFS_PTR_CTREXPEL: Flush any data to the printer, then throw the media out. WFS_PTR_CTRLRETRACT: Flush any data to the printer, then retract the media.
dwPaperSource	Specifies the Paper source to use when printing this form: it must be WFS_PTR_PAPERANY or WFS_PTR_PAPERUPPER.

[Output parameters] None.

[Specific error codes]

WFS_ERR_PTR_FILENOTFOUND	The specified file cannot be found.
WFS_ERR_PTR_FILEIOERROR	The specified file cannot be read.
WFS_ERR_PTR_SOURCEINVALID	The selected paper source is not supported by the hardware.
WFS_ERR_PTR_PAPEROUT	The paper supply is empty.

Same error codes of command WFS_CMD_PTR_CONTROL_MEDIA can be generated too.
Note that a WFS_ERR_NOMEDIAPRESENT event can be generated too, if the dwMediaControl flag WFS_PTR_CTRLRETRACT is set (retract after print).

[Specific events]

WFS_SRVE_PTR_MEDIATAKEN	The media has been taken by the user
WFS_USRE_PTR_PAPERTHRESHOLD	The paper supply is low or empty; operator intervention is required. Note that this event is sent only once, at the point at which the supply becomes low or empty. It is sent with WFS_PTR_PAPERLOW or WFS_PTR_PAPEROUT status.

Same [Specific events] of command WFS_CMD_PTR_CONTROL_MEDIA can be generated too.

WFS_CMD_PTR_LOAD_DEFINITION

[Input parameters]

```
LPWFSPTRLOADDEFINITION lpLoadDefinition
        typedef struct _wfs_ptr_load_definition {
        LPSTR lpszFileName;
        BOOL bOverwrite;
} WFSPTRLOADDEFINITION, *LPWFSPTRLOADDEFINITION;
```

Where:

PARAMETER	MEANING
lpszFileName	Pointer to the null-terminated file name Full path and file name of the file to be loaded. Only the first form or media definition in the file will be loaded.
bOverwrite	This flag specifies if an existing dorm or media with the same name is to be replaced. When more services share the XFSMEDIA or XFSFORM folder, the specified form or media will be added and made permanent for all the printers using these folders.

[Output parameters] None.

[Specific error codes]

WFS_ERR_PTR_FILENOTFOUND	The specified file cannot be found.
WFS_ERR_PTR_FORMINVALID	The form is invalid.
WFS_ERR_PTR_MEDIAINVALID	The media definition is invalid.
WFS_ERR_PTR_DEFINITIONEXISTS	The specified form or media definition already exists and the bOverwrite flag was FALSE.

[Specific events]

WFS_SRVE_PTR_DEFINITIONLOADED	A form or media definition has been loaded; an existing definition may have been modified by replacement.

WFS_CMD_PTR_SUPPLY_REPLENISH

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_POWER_SAVE_CONTROL

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

WFS_CMD_PTR_CONTROL_PASSBOOK

This command is not implemented.

Immediate result: WFS_ERR_UNSUPP_COMMAND

EVENTS

WFS_SYSE_HARDWARE_ERROR, WFS_SYSE_SOFTWARE_ERROR

These events are sent immediately before a WFS_ERR_HARDWARE_ERROR (or device class specific error code) or WFS_ERR_SOFTWARE_ERROR error code that is returned as the command completion in order to report more information about the error.

[Event associated data]

```
typedef struct _wfs_hwerror {
        LPSTR lpszLogicalName;
        LPSTR lpszPhysicalName;
        LPSTR lpszWorkstationName;
        LPSTR lpszAppID;
        DWORD dwAction;
        DWORD dwSize;
        LPBYTE lpbDescription;
} WFSHWERROR, * LPWFSHWERROR;
```

Where:

PARAMETER	MEANING	
lpszLogicalName	Pointer to the logical name of the service.	
lpszPhysicalName	Pointer to the string "Custom_PTR_XFS"	
lpszWorkstationName	Pointer to the workstation name where the Service Provider is running.	
lpszAppID	Pointer to the application ID name associated with the session that generated the error.	
dwAction	WFS_ERR_ACT_NOACTION	No action required. Error was autorecovered.
	WFS_ERR_ACT_RESET	Reset device to attempt recovery.
	WFS_ERR_ACT_SWERROR	A software error occurred. Contact software vendor.
	WFS_ERR_ACT_CONFIG	A configuration error occurred. Check configuration.
	WFS_ERR_ACT_HWCLEAR	Recovery is not possible. A manual intervention for clearing the device is required. This value is only used for hardware errors.
	WFS_ERR_ACT_HWMAINT	Recovery is not possible. A technical maintenance intervention is required. This value is only used for hardware errors.
dwSize	The size in bytes of the string pointed by the lpbDescription pointer.	
lpbDescription	A zero terminated ASCII string describing the error.	

WFS_SYSE_DEVICE_STATUS

Status changes of logical services are reported by this event. This is in addition to being reported by the WFS_INF_PTR_STATUS info command.

[Event associated data]

```
typedef struct _wfs_devstatus {
        LPSTR lpszPhysicalName;
        LPSTR lpszWorkstationName;
        DWORD dwState;
} WFSDEVSTATUS, * LPWFSDEVSTATUS;
```

Where:

PARAMETER		MEANING	
lpszPhysicalName		equal to "Custom_PTR_XFS"	
lpszWorkstationName		Name of the workstation where the Service Provider is running.	
dwState		Specifies the new state of the physical device managed by the service as one of the following:	
	fwDevice	WFS_PTR_DEVONLINE	The device is online.
		WFS_PTR_DEVOFFLINE	The device is powered off or not connected.
		WFS_PTR_DEVHWERROR	The device is inoperable.
		WFS_PTR_DEVUSERERROR	The device is inoperable because the printer cover is open.

WFS_USRE_PTR_RETRACTBINTHRESHOLD

[Event associated data] *typedef struct _wfs_ptr_bin_threshold {*

```
typedef struct _wfs_ptr_bin_threshold {
            USHORT usBinNumber;
            WORD wRetractBin;
} WFSPTRBINTHRESHOLD, *LPWFSPTRBINTHRESHOLD;
```

Where:

PARAMETER	MEANING	
usBinNumber	It is always 1.	
wRetractBin	WFS_PTR_RETRACTBINOK	The retract bin is empty.
	WFS_PTR_RETRACTBINHIGH	The retract bin has reached its nearly full threshold.
	WFS_PTR_RETRACTBINFULL	The retract bin has reached its full threshold.

WFS_SRVE_PTR_MEDIATAKEN

[Event associated data]   None.

WFS_USRE_PTR_PAPERTHRESHOLD

[Event associated data]

```
typedef struct _wfs_ptr_paper_threshold {
            WORD wPaperSource;
            WORD wPaperThreshold;
} WFSPTRPAPERTHRESHOLD, *LPWFSPTRPAPERTHRESHOLD;
```

Where:

PARAMETER	MEANING	
wPaperSource	It is WFS_PTR_PAPERUPPER.	
wPaperThreshold	WFS_PTR_PAPEROUT	The paper supply is empty.
	WFS_PTR_PAPERLOW	The paper supply is low.
	WFS_PTR_PAPERFULL	The paper supply is full.

WFS_EXEE_PTR_MEDIAPRESENTED

[Event associated data]

```
typedef struct _wfs_ptr_media_presented {
        USHORT usWadIndex;
        USHORT usTotalWads;
} WFSPTRMEDIAPRESENTED, *LPWFSPTRMEDIAPRESENTED;
```

WFS_SRVE_PTR_MEDIADETECTED

[Event associated data]

```
typedef struct _wfs_ptr_media_detected {
        WORD wPosition;
        USHORT usRetractBinNumber;
} WFSPTRMEDIADETECTED, *LPWFSPTRMEDIADETECTED;
```

Where:

PARAMETER	MEANING	
wPosition	WFS_PTR_PAPEROUT WFS_PTR_PAPERLOW WFS_PTR_PAPERFULL	The paper supply is empty. The paper supply is low. The paper supply is full.
usRetractBinNumber	It is always 1.	

WFS_EXEE_PTR_FIELDERROR

[Event associated data]

```
typedef struct _wfs_ptr_field_failure {
        LPSTR lpszFormName;
        LPSTR lpszFieldName;
        WORD wFailure;
} WFSPTRFIELDFAIL, *LPWFSPTRFIELDFAIL;
```

Where:

PARAMETER	MEANING	
lpszFormName	Points to the null-terminated form name.	
lpszFieldName	Points to the null-terminated field name.	
wFailure	WFS_PTR_FIELDREQUIRED	The specified field must be supplied by the application.
	WFS_PTR_FIELDSTATICOVWR	The specified field is static and thus cannot be overwritten by the application.
	WFS_PTR_FIELDNOTFOUND	The specified field does not exist.

WFS_SRVE_PTR_DEFINITIONLOADED

[Event associated data] *typedef struct _wfs_ptr_definition_loaded {*
LPSTR lpszDefinitionName;
DWORD dwDefinitionType;
*} WFSPTRDEFINITIONLOADED, *LPWFSPTRDEFINITIONLOADED;*

Where:

PARAMETER	MEANING	
lpszDefinitionName	Specifies the name of the just loaded form or media.	
dwDefinitionType	WFS_PTR_FORMLOADED	The form identified by lpszDefinitionName has been loaded.
	WFS_PTR_MEDIALOADED	The media identified by lpszDefinitionName has been loaded.

Code : 00577200M000076 Rev. 1.00

CUSTOM S.p.A.
World Headquarters
Via Berettine, 2/B - 43010 Fontevivo, Parma ITALY
Tel. +39 0521 680111 - Fax +39 0521 610701
info@custom.biz - www.custom.biz