FSG Open Printing – Print Channel Manager

Print Channel Manager Standard

Version 0.8

June 6, 2006

Editors (alphabetical)

Authors	***Company***	***E-Mail***
Kentaro Ide	Seiko Epson	Ide.Kentaro@exc.epson.co.jp

Authors (alphabetical)

Authors	***Company***	***E-Mail***
Kazuo Nomura	Seiko Epson	Nomura.Kazuo@exc.epson.co.jp
Kentaro Ide	Seiko Epson	Ide.Kentaro@exc.epson.co.jp

Contributors (alphabetical)

Authors	***Company***	***E-Mail***
Glen Petrie	Seiko Epson Corp.	gpetrie@eitc.epson.co.jp
Koji Otani	AXE, Inc.	sho@axe-inc.co.jp
Osamu Mihara	Fuji Xerox Printing Systems Co., Ltd.	mihara.osamu@fxpsc.co.jp
Shimpei Kitayama	Epson Avasys Corp.	shimpei.kitayama@avasys.jp
Yasumasa Toratani	Canon Inc.	toratani.yasumasa@canon.co.jp

FSG Open Printing – Print Channel Manager
Copyright Notice

Table of Contents

Introduction

This document describes the Print Channel Manager, also known as PCM. PCM will be wrap difference of physical interface (USB, Parallel direct, 1284.4 etc.) and offer standard interface to Status Monitor, Driver and Capability.

Background

Today, Linux environment have many physical printer or peripheral interface like USB, Parallel direct, IEEE 1284.4 etc. On the other hand, OpenPrinting is for the purpose of that offers the generic interface to upper class layer. Therefore, we need specify the PCM layer that wraps difference of physical interface.

Terminology and Acronyms

Conformance Terminology

In this document, the uppercase terms “MUST”, “MUST NOT”, “REQUIRED”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are intended to be interpreted as described in [RFC2119]

In this document, each conformance statement uses one of the terms:

Term	***Meaning***
MUST	Implementation supchannel is REQUIRED for conformance to this specification.
SHOULD	Implementation supchannel is RECOMMENDED for conformance to this specification.
SHOULD NOT	Implementation supchannel is NOT RECOMMENDED for conformance to this specification.
MAY	Implementation supchannel is OPTIONAL for conformance to this specification.

In this document, each operation, object, or attribute is defined as:

Term	***Meaning***
REQUIRED	Each implementation MUST supchannel object operations or attributes.
RECOMMENDED	Each implementation SHOULD supchannel object operations for interoperability
OPTIONAL	Each implementation MAY supchannel object operations or attributes.

Other Terminology

Miscellaneous Terms:

Term	***Meaning***

Term	*Meaning*
IN	Input data to function
OUT	Output data from function
PCM Plug-in	PCM Plug-in is plug-in module with vender specific code.

Acronyms

Acronyms:

Acronyms	*Meaning*	*Source*
FSG	Free Standards Group	http://www.freestandardsgroup.org/
FSG/OP	Free Standards Group – Open Printing	http://www.openprinting.org/
PCM	Print Channel Manager	
SM	Status Monitor	
MFP	Multi Function Printer	

Print Channel Manager Model

Architecture Overview

PCM Architecture

PCM is a wrapper modules that wrapping physical layer of printer device include MFP.
PCM provide communication channel interface to upper layer like SM, Driver or Capability. When opening channel, the upper layer module specify device type, Plug-in, device node and other information for printing with printer ID.PCM relays communication with a device and a upper layer, after starting the module which changes the communications protocol according to a device and a printer and securing a communication course by specified printer ID. The communication method between PCM and a communications protocol conversion module, PCM Plug-in, is defined as Plug-in Interface. After securing a communication path, a upper layer read or write data using Read or Write function, uses Close function after a reading-and-writing end, and closes a port. Exception handling, such as cancellation processing in the middle of printing, uses Control function.

Requirements

Define what requirements to PCM and Plug-in are as follow.

• Requirements to PCM

- Under the Open Printing System's architecture, Open Printing module will be using PCM for communicate to device like a printer. Upper layer can getting status data without print job.
- PCM provide only port control feature but do not provide Job control feature.
- PCM use PCM Plug-in for wrapping communication protocol or vendor specific procedure. Plug-in specified by the upper layer of PCM.PCM supports not only single function printer but multi function printer.
- PCMはMFPが持つ機能のうち、最低でも印刷機能についてはサポートする。印刷以外の機能については、将来的にサポートされる。

Requirements to Plug-in

- Plug-in communicates by using the protocol corresponding to the device.
- Plug-in changes into a corresponding protocol the printing data received via PCM API, and sends it out to Printer device.
- According to the demand via PCM API, Plug-in reads data from device

and returns it to PCM.

- Plug-in changes into a corresponding protocol the control command received via PCM API, and sends it out to device. Plug-in returns to PCM the status data read in device. Timing control of the reading and writing of data to device and initialization processing of a printer are processed in the form closed inside Plug-in.

API

fsgpcmOpen

Name

fsgpcmOpen – Open a channel to the printer

Synopsis

FSGPCM_Ctx *fsgpcmOpen (FSGPCM_CHAR *fsgDeviceURI,
FSGPCM_CHAR *fsgpcmMode, int iFSGPCMIOError);

Arguments

argument	***IN/OUT***	***description***
fsgpcmDeviceURI	IN	Device URI for specify the information of device that actually binded to the system. NOTE: In present, this parameter is defined as URI but not ID. Result of Printing-architecture sub group's discussion will be reflected in this parameter in future. /* Printer ID for specify the registry information of printer device that actually binded to the system. Registry information have information that specify the printer vendor, model, PCM Plug-in, device node, etc... */
fsgpcmMode	IN	Type of access permitted
iFSGPCMIOError	OUT	Error number. If error occurred when open device, this function will return error number.

Description

This function is used in order to open the communication channel to device.
In case Communication channel is opened using this function, the target channel classification, the device path, and plug-in are specified by Device URI.A upper layer is able for channel opened by this function to have been independent for every channel classification specified by

Device URI, and to access simultaneously to each channel.
Format of Device URI is as below.

pcm://*host*:*port*/*DeviceName*?devtype=*DeviceType*&devnode=*DeviceNode*&model=*DeviceModelName*&vendor=*DeviceVendor*&plugin=*DevicePlugin*

Name	***Requirement***	***Description***
host	SHOULD	Host name or IP address of machine that work PCM The value of localhost is used when it omits.
port	SHOULD	Port used by PCM A default port is used when it omits.
DeviceName	MUST	Unique ID of each device
DeviceType	MUST	Device type like a printer, scanner or something
DeviceNode	SHOULD	Device node that printer connected e.x. /dev/ulpt0
DeviceModelName	SHOULD	Model name of device
DeviceVendor	SHOULD	Vendor name of device
DevicePlugin	MUST	Plug-in name or file name

/*
Comment notes - Printer ID and registry will be defined in discussion of "Open Printing Architecture" sub WG. PCM and other open printing system will be according to result of discussion of architecture group.
-- comment by Kentaro Ide, 2005/11/4
Additional comment -
PCM author team decided to change "Printer ID" to "Device URI" because Printer ID is not defined in Open Printing System yet.
-- comment by Kentaro Ide, 2006/6/15
*/

The character string fsgpcmMode specifies the type of access.
Detail of mode is as follows.

Characters for specify the direction mode.

"r"
Open device for reading data.
"w"
Open device for writing data.

Characters for specify the translation mode.
"t"
Open in text mode.

"b"

Open in binary mode.

Direction mode character and translation mode character will be using with combinations.

Return Value

Handle to channel or NULL when error.

fsgpcmClose

Name

fsgpcmClose – Close the channel to the printer

Synopsis

int fsgpcmClose (FSGPCMCtx *hFSGPCMCtx);

Arguments

Argument	IN/OUT	Description
hFSGPCMCtx	IN	Handle to channel want to close

Description

This function is used in order to close channel opened by fsgpcmOpen.The port specified by hFSGPCMCtx is closed.

Return Value

Value	Description
FSGPCM_OK	Normal end
FSGPCM_ERROR	Error

fsgpcmRead

Name

fsgpcmRead – Read data stream from the channel.

Synopsis

size_t fsgpcmRead (void *pBuf, size_t nBufSize, size_t nCount, FSGPCM_Ctx *hFSGPCMCtx);

Arguments

Argument	IN/OUT	Description

Argument	*IN/OUT*	*Description*
pBuf	IN	Pointer to buffer for reading data.
nBufSize	IN	Buffer size of element
nCount	IN	Number of nBufSize elements
hFSGPCMCtx	IN	Handle to channel

Description

This function is used in order to read data from the channel opened by fsgpcmOpen. This function reads, into the array pointed to by pBuf, up to nCount elements whose size is specified by nBufSize, from the channel pointed to by hFSGPCMCtx.

Return Value

Number of actually read data elements.
In case of error or end-of-file, return value will be less than nCount.

fsgpcmWrite

Name

fsgpcmWrite – Write data stream to the channel.

Synopsis

size_t fsgpcmWrite (void *pBuf, size_t nBufSize, size_t nCount, FSGPCMCtx *hFSGPCMCtx);

Arguments

Argument	*IN/OUT*	*Description*
pBuf	IN	Pointer to data write buffer
nBufSize	IN	Buffer size of element
nCount	IN	Number of nBufSize elements
hFSGPCMCtx	IN	Handle to data write channel

Description

This function is used in order to write in data to channel opened by fsgpcmOpen.The data is wrote from buffer pointed by pBuf to the channel pointed by hFSGPCMCtx. Data is repeatedly written the number of times specified by nCount in the number of bytes specified by nBufSize.

Return Value

Number of actually wrote data elements.

In case of error, return value will be less than nCount.

fsgpcmIOError

Name

fsgpcmIOError – Get a error number

Synopsis

int fsgpcmIOError (FSGPCMCtx hFSGPCMCtx);

Arguments

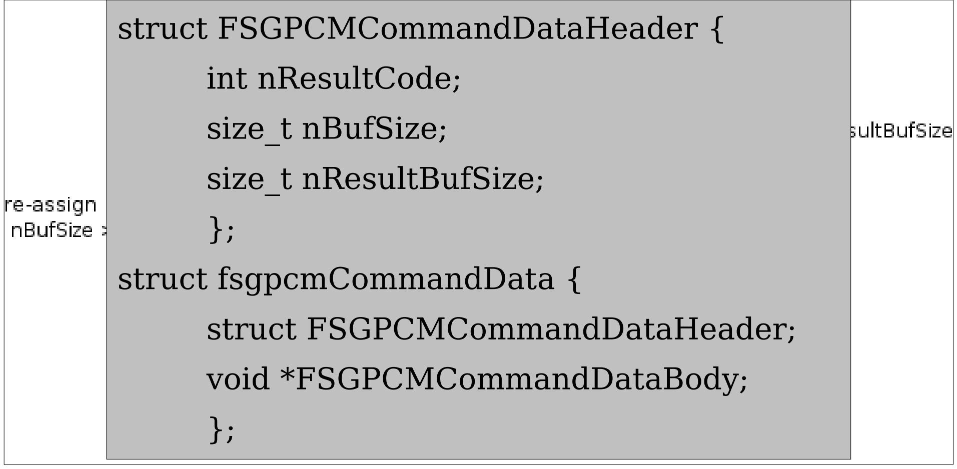
Argument	IN/OUT	Description
hFSGPCMCtx	IN	Handle to Control channel

Description

This function is used for get a error number when error occurred.

Return Value

Value	Description
FSGPCM_ERROR_PARAM	Invalid parameters This error will be returned when using unsupported parameter.
FSGPCM_ERROR_NOSUPPORT	Invalid control command This error will be returned when using un-supported control command.
FSGPCM_ERROR_NOCONNECT	Error on connection with file descriptor *Fatal* This error will be returned when can't read/write to File Descriptor (FD) with something reason include protocol mismatch.
FSGPCM_ERROR_NOMEM	Can't allocate memory.
FSGPCM_ERROR_SIGNAL	Interrupted by signal
FSGPCM_ERROR_TIMEOUT	Timeout If set NO_BLOCK as timeout parameter, PCM_ERROR_TIMEOUT return when Plug-in is not ready (BUSY).
FSGPCM_ERROR_VERSION	Unsupported version. In the case of newer version of Plug-in API don't have backward compatibility, PCM return this error when using differing version Plug-in.
FSGPCM_ERROR_UNKNOWN	Unknown error. All undefined error.



```
struct FSGPCMCommandDataHeader {
        int nResultCode;
        size_t nBufSize;
        size_t nResultBufSize;
        };
struct fsgpcmCommandData {
        struct FSGPCMCommandDataHeader;
        void *FSGPCMCommandDataBody;
        };
```

fsgpcmIOControl

Name

fsgpcmIOControl – send control command to device

Synopsis

FSGPCM_Result *fsgpcmIOControl (FSGPCMCtx hFSGPCMCtx, int iFSGPCMCommand, void *pFSGPCMArg);

Arguments

Argument	*IN/OUT*	*Description*
hFSGPCMCtx	IN	Handle to Control channel
iFSGPCMCommand	IN	Control command
pFSGPCMArg	IN	Pointer to control description to device

Description

/* 要チェック Since control command is sent out to Control channel specified by hFSGPCMCtx, this function is used.
The type of control is specified as “Control Command”. A upper layer can perform detailed control to device by control data handed over using pFSGPCMArg.
The information from the device side expected by publishing command is acquired in the form of result data via FSGPCM_Result.
In case of GET_STATUS, status data read sequence is as follow.

control data/result data is a structure as follow.
*/
*** variable name, type and result code are T.B.D.**

Name	*IN/OUT*	*Description*
FSGPCMCommandDataHeader	IN/OUT	Header of data set
FSGPCMCommandDataBody	IN/OUT	Content of control data
nResultCode	OUT	Result code that indicate like a error type
nBufSize	IN	Size of buffer that PCM allocated

Name	***IN/OUT***	***Description***
nResultBufSize	OUT	Size of buffer that PCM Plug-in allocated

Result Code	***Description***
FSGPCM_OK	Normal End
FSGPCM_ERROR	Can't finish the get/send process

Return Value

Control/result data structure

Control Command

Control Command which should be outputted by Control Channel is defined as follow.

Command	***Device***	***Category***	***Description***
RESET	Printer Scanner	Control	Reset device
VENDOR_SPECIFIC	ALL	All	Vendor specific command

Constants

Return Values (T.B.D)

FSGPCM_OK	
FSGPCM_ERROR	
FSGPCM_NOSUPPORT	
FSGPCM_CANCELED	

Plug-in API

fsgpcpNew

Name

fsgpcpNew – Create new object with Plug-in

Synopsis

FSGPCP_Ctx *fsgpcpNew (FSGPCM_CHAR *fsgDeviceURI, char FSGPCPVersion, int iFSGPCPIOError);

Arguments

argument	IN/OUT	description
fsgDeviceURI	IN	Device URI for specify the information of device that actually binded to the system. Reffer fsgpcmOpen for detail.
FSGPCP_Version	OUT	Version number. Return version number of Plug-in module.
iFSGPCPIOError	OUT	Error number. If error occurred when open device, this function will return error number.

Description

This function is used in order to create Plug-in object for open channel to the device.
When already created Plug-in object before call this function, this function return handle to existed object.

Return Value

FSGPCP_Ctx *hFSGPCPCtx
Handle to channel or NULL when error.

hFSGPCPCtx->Open

Name

hFSGPCPCtx->Open – Open a channel to the device

Synopsis

FSGPCP_Ctx hFSGPCPCtx->Open (FSGPCP_Ctx *hFSGPCPCtx, FSGPCM_CHAR *fsgpcpMode, int iFSGPCPIOError);

Arguments

argument	IN/OUT	description
hFSGPCPCtx	IN	Handle to Plug-in object
fsgpcpMode	IN	Type of access permitted
iFSGPCPIOError	OUT	Error number. If error occurred when open device, this function will return error number.

Description

This function is used in order to open the communication channel to device.

The character string fsgpcpMode specifies the type of access.
Detail of mode is as follows.

Characters for specify the direction mode.

“r”
Open device for reading data.
“w”
Open device for writing data.

Characters for specify the translation mode.
“t”
Open in text mode.
“b”
Open in binary mode.

Direction mode character and translation mode character will be using with combinations.

Return Value

FSGPCP_Ctx *hFSGPCPCHANELLCtx
Handle to channel or NULL when error.

hFSGPCPCtx->Close

Name

fsgpcpClose – Close the channel to the printer

Synopsis

int fsgpcpClose (FSGPCP_Ctx   *hFSGPCPCHANNELCtx);

Arguments

Argument	IN/OUT	Description
hFSGPCPCHANNELCtx	IN	Handle to channel want to close

Description

This function is used in order to close channel opened by fsgpcpOpen.
The port specified by hFSGPCPCHANELLCtx is closed.

Return Value

	Description
FSGPCM_OK	Normal end
FSGPCM_ERROR	Error

fsgpcpDestroy

Name

fsgpcpDestroy – Destroy the channel object

Synopsis

```
int fsgpcpClose ( FSGPCP_Ctx   *hFSGPCPCtx );
```

Arguments

Argument	IN/OUT	Description
hFSGPCPCtx	IN	Handle to Plug-in object want to close

Description

This function is used in order to destroy Plug-in object created by fsgpcpNew. If the count of fsgpcpNew call is larger than count of fsgpcpDestroy, PCM not really destroy Plug-in object but function will be ending without error.

Return Value

	Description
FSGPCM_OK	Normal end
FSGPCM_ERROR	Error

hFSGPCPCHANNELCtx->hFSGPCPCtx->Read

Name

hFSGPCPCHANNELCtx->hFSGPCPCtx->Read – Read data stream from the channel.

Synopsis

size_t hFSGPCPCHANNELCtx->hFSGPCPCtx->Read (void *pBuf, size_t nBufSize, size_t nCount, FSGPCP_Ctx *hFSGPCPCHANNELCtx);

Arguments

Argument	*IN/OUT*	*Description*
pBuf	IN	Pointer to buffer for reading data.
nBufSize	IN	Buffer size of element
nCount	IN	Number of nBufSize elements
hFSGPCPCHANNELCtx	IN	Handle to channel

Description

This function is used in order to read data from the channel opened by hFSGPCPCtx->Open.
This function reads, into the array pointed to by pBuf, up to nCount elements whose size is specified by nBufSize, from the channel pointed to by hFSGPCPCHANNELCtx.

Return Value

Number of actually read data elements.
In case of error or end-of-file, return value will be less than nCount.

hFSGPCPCHANNELCtx->hFSGPCPCtx->Write

Name

hFSGPCPCHANNELCtx->hFSGPCPCtx->Write – Write data stream to the channel.

Synopsis

size_t hFSGPCPCHANNELCtx->hFSGPCPCtx->Write (void *pBuf, size_t nBufSize, size_t nCount, FSGPCP_Ctx *hFSGPCPCHANNELCtx);

Arguments

Argument	*IN/OUT*	*Description*
pBuf	IN	Pointer to data write buffer
nBufSize	IN	Buffer size of element

Argument	*IN/OUT*	*Description*
nCount	IN	Number of nBufSize elements
hFSGPCMCHANNELCtx	IN	Handle to data write channel

Description

This function is used in order to write in data to channel opened by hFSGPCPCtx->Open. The data is wrote from buffer pointed by pBuf to the channel pointed by hFSGPCPCHANNELCtx. Data is repeatedly written the number of times specified by nCount in the number of bytes specified by nBufSize.

Return Value

Number of actually wrote data elements.
In case of error occurred, return value will be less than nCount.

hFSGPCPCHANNELCtx->hFSGPCPCtx->IOError

Name

hFSGPCPCHANNELCtx->hFSGPCPCtx->IOError – Get a error number

Synopsis

int hFSGPCPCHANNELCtx->hFSGPCPCtx->IOError (FSGPCP_Ctx hFSGPCPCtx);

Arguments

Argument	*IN/OUT*	*Description*
hFSGPCPCtx	IN	Handle to Control channel

Description

This function is used for get a error number when error occurred.

Return Value

Value	*Description*
FSGPCM_ERROR_PARAM	Invalid parameters This error will be returned when using unsupported parameter.
FSGPCM_ERROR_NOSUPPORT	Invalid control command This error will be returned when using un-supported control command.

Value	*Description*
FSGPCM_ERROR_NOCONNECT	Error on connection with file descriptor *Fatal* This error will be returned when can't read/write to File Descriptor (FD) with something reason include protocol mismatch.
FSGPCM_ERROR_NOMEM	Can't allocate memory.
FSGPCM_ERROR_SIGNAL	Interrupted by signal
FSGPCM_ERROR_TIMEOUT	Timeout If set NO_BLOCK as timeout parameter, PCM_ERROR_TIMEOUT return when Plug-in is not ready (BUSY).
FSGPCM_ERROR_VERSION	Unsupported version. In the case of newer version of Plug-in API don't have backward compatibility, PCM return this error when using differing version Plug-in.
FSGPCM_ERROR_UNKNOWN	Unknown error. All undefined error.

hFSGPCPCHANNELCtx->hFSGPCPCtx->IOControl

Name

hFSGPCPCHANNELCtx->hFSGPCPCtx->IOControl – send control command to device

Synopsis

FSGPCM_Result hFSGPCPCHANNELCtx->hFSGPCPCtx->IOControl (FSGPCP_Ctx hFSGPCPCtx, int iFSGPCPCommand, void *pFSGPCPArg);

Arguments

Argument	*IN/OUT*	*Description*
hFSGPCPCtx	IN	Handle to Control channel
iFSGPCPCommand	IN	Control command
pFSGPCPArg	IN	Pointer to control description to device

Description

Since control command is sent out to Control channel specified by hFSGPCPCtx, this function is used.
The type of control is specified as “Control Command”. A upper layer can perform detailed control to device by control data handed over using pFSGPCPArg.

The information from the device side expected by publishing command is acquired in the form of result data via FSGPCPResult.
In case of GET_STATUS, status data read sequence is as follow.

Name	***IN/OUT***	***Description***
FSGPCMCommandDataHeader	IN/OUT	Header of data set
FSGPCMCommandDataBody	IN/OUT	Content of control data
nResultCode	OUT	Result code that indicate like a error type
nBufSize	IN	Size of buffer that PCM allocated
nResultBufSize	OUT	Size of buffer that PCM Plug-in allocated

Result Code	***Description***
FSGPCM_OK	Normal End
FSGPCM_ERROR	Can't finish the get/send process

Return Value

Control/result data structure

APPENDIX

CHANGES

Date	***Affected Version***	***Editor***	***Change***
2005/07/10	0.1	Kentaro Ide	First public draft
2005/08/30	0.3	Kentaro Ide	- Change detail of functions as below. fsgpcmOpen, fsgpcmClose, fsgpcmRead, fsgpcmWrite - Change function name fsgpcmControl > fsgpcmIOControl
2005/11/4	0.7	Kentaro Ide	- Change license from GFDL to MIT - Add function fsgpcmIOError
2006/06/16	0.8	Kentaro Ide	- Add Error code - Add Plug-in API - Change “Printer ID” to “DeviceURI”