OpenPrinting WG Bi-di Plug-in API

TORATANI Yasumasa
<toratani.yasumasa@canon.co.jp>
OpenPrinting WG Japan/Asia
Canon Inc.

2004-3-23, 24, 25

Agenda

- Introduction
  - What is the Bi-di Plug-in API?
  - Architecture Overview
- API Detail
- Corresponding API
  - Bi-di Plug-in Shared Library Interface
  - Bi-di Plug-in Process Interface
- Printer Status Data Format
- Issues/Concerns, Next and Info.

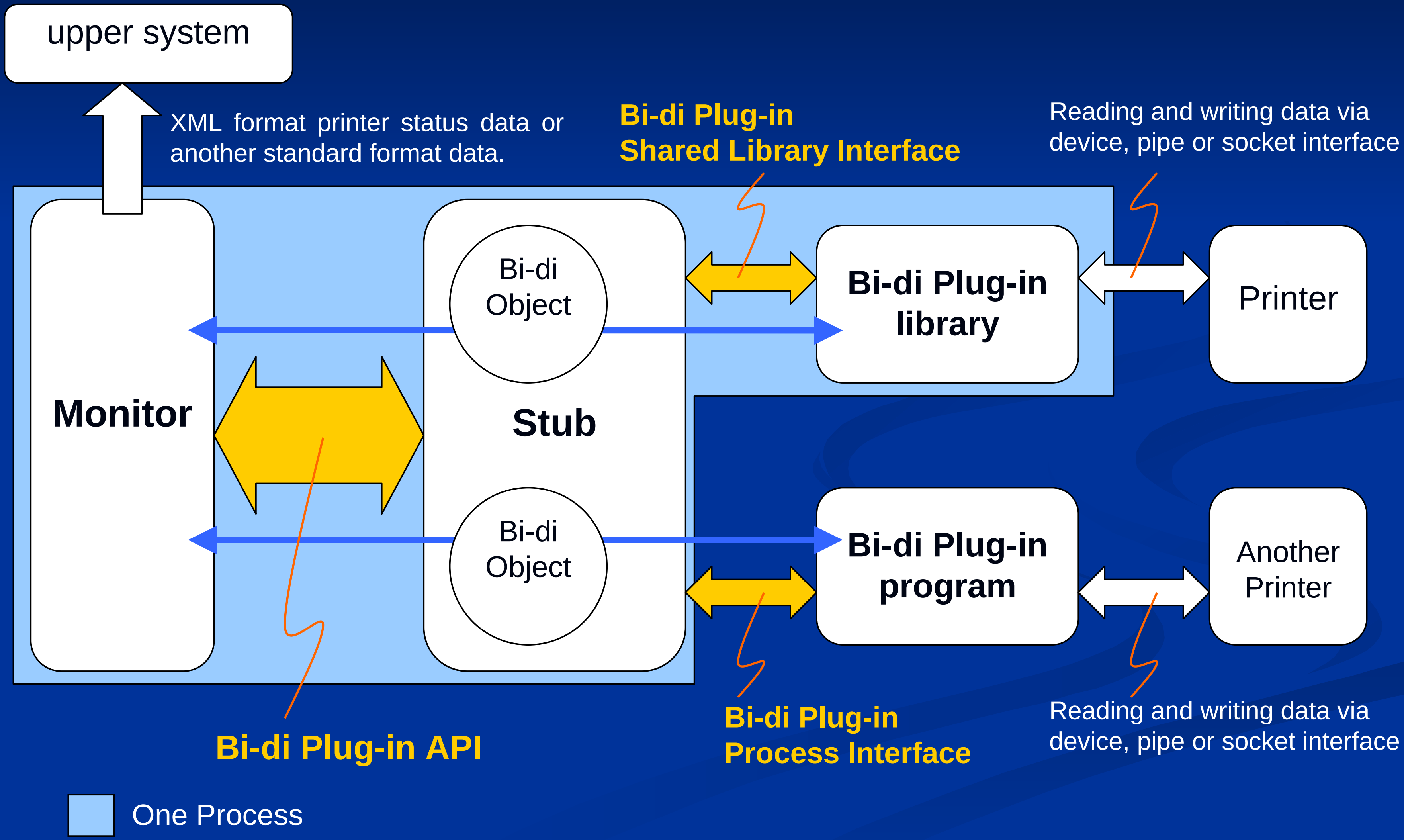
Introduction

What is the Bi-di Plug-in API ?

- Bi-di Plug-in API is:
  - An interface for obtaining the printer status and/or information from the printer as well as for sending the printing data to the printer.

- Bi-di Plug-in API contains functions to:
  - Create/delete a Bi-di Object.
  - Obtain the Bi-di Plug-in capabilities.
  - Control printing jobs.
  - Obtain the printer status from a Bi-di Plug-in module.
  - Send the printing data to a Bi-di Plug-in module.
  - Deal with a miscellaneous I/O control.

Architecture Overview



upper system

XML format printer status data or another standard format data.

Bi-di Plug-in
Shared Library Interface

Reading and writing data via device, pipe or socket interface

Monitor

Bi-di Object

Stub

Bi-di Object

Bi-di Plug-in library

Printer

Bi-di Plug-in program

Another Printer

Bi-di Plug-in API

Bi-di Plug-in
Process Interface

Reading and writing data via device, pipe or socket interface

🟦 One Process

Monitor and Stub

- “Monitor” is:
  - A program which calls the Bi-di Plug-in API entries provided by the Stub.
    - Applications / middlewares
    - CUPS backends
    - Printer driver which needs the printer status/information.
    - etc..
- “Stub” is:
  - A library which connects the Monitor and the Bi-di Plug-in module, and provides the Bi-di Plug-in API entries to the Monitor.
    - Shared library or Static library.

Bi-di Plug-in module

- "Bi-di Plug-in module" is:
  - A shared library or an individual program which provides a set of the Bi-di Plug-in API functions.
- "Bi-di Plug-in library" is:
  - A Bi-di Plug-in module as a shared library being linked by the Stub.
    - Run as the same process as the Stub and Monitor.
      → Plug-in obtains the printer status/information **passively**.
      → Plug-in can hook signals of the Monitor.
- "Bi-di Plug-in program" is:
  - A Bi-di Plug-in module as an individual program which communicates with the Stub via an inter-process communication interface.
    - Obtain the printer status individually from the Stub and Monitor.
      → Plug-in can obtain the printer status/information **actively**.
      → Plug-in can deal with signals individually from the Monitor.

Corresponding API

- Bi-di Plug-in Shared Library Interface
  - An interface between the Stub and the Bi-di Plug-in module.
- Bi-di Plug-in Process Interface
  - An interface between the Stub and the Bi-di Plug-in program.

API Detail

Bi-di Plug-in API

- Create a Bi-di Object <MUST be supported>

Synopsis

```
BidiC *bidiNew(char *pName, int fdRead, int fdWrite,
   char *pURI);
```

Arguments

pName – Name of the Bi-di Plug-in library or Bi-di Plug-in program.

fdRead – File descriptor for reading the printer status data from the Bi-di Plug-in module.

fdWrite – File descriptor for writing the printer command data to the Bi-di Plug-in module.

pURI – Device URI.

Bi-di Plug-in API

- Delete a Bi-di Object <MUST be supported>

Synopsis

```
void bidiDestroy(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Obtain the Bi-di Plug-in module capabilities
  <MUST be supported>

Synopsis

```
int bidiGetGap(BidiC *pBidiC, BidiCap cap);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

cap – Enum of the Bi-di Object capabilities.

BIDI_CAP_WRITE	Supports writing functions.
BIDI_CAP_JOB	Supports job functions.
BIDI_CAP_CTRL	Supports bidiCtrl() function.

Bi-di Plug-in API

- Declare the start of the printing job sequence

Synopsis

```
int bidiStartJob(BidiC *pBidiC, int idJob);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

idJob – Job ID.

Bi-di Plug-in API

- Declare the end of the printing job sequence

Synopsis

```
int bidiEndJob(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Declare the cancellation of the printing job sequence

Synopsis

```
int bidiCancelJob(BidiC *pBidiC, int idJob);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

idJob – Job ID.

Bi-di Plug-in API

- Get the file descriptor for reading the printer status
  <MUST be supported>

Synopsis

```
int bidiGetReadFD(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Start the printer status reading sequence

  <MUST be supported>

Synopsis

```
int bidiStartRead(BidiC *pBidiC, BidiReadMode
  idReadMode, char *pLang);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

idReadMode – Enum of the Bi-di Object reading mode.

BIDI_READ_PRT_MIB_ALL	Read all the printer status data.
BIDI_READ_PRT_MIB_SUMMARY	Read the summarized status data.

pLang – Pointer to the language string.

e.g.) ja_JP.UTF-8

Bi-di Plug-in API

- Read the printer status
  <MUST be supported>

Synopsis

```
int bidiRead(BidiC *pBidiC, void *pBuf, int nBufBytes);
```

Arguments

pBidiC – Pointer to the Bi-di Object.
pBuf – Pointer to the buffer for storing the printer status data.
nBufBytes – Number of bytes for reading the printer status data.

Bi-di Plug-in API

- Stop the printer status reading sequence
  <MUST be supported>

Synopsis

```
int bidiEndRead(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Get the file descriptor for writing the printing data

Synopsis

```
int bidiGetWriteFD(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Start the printing data writing sequence

Synopsis

```
int bidiStartWrite(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Write the printing data

Synopsis

```
int bidiWrite(BidiC *pBidiC, void *pBuf, int nBufBytes);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

pBuf – Pointer to the buffer for writing data.

nBufBytes – Number of bytes of the writing data.

Bi-di Plug-in API

- Stop the printing data writing sequence

Synopsis

```
int bidiEndWrite(BidiC *pBidiC);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

Bi-di Plug-in API

- Deal with a miscellaneous I/O control

Synopsis

```
int bidiCtrl(BidiC *pBidiC, int idRequest, void *pData,
   int nDataBytes);
```

Arguments

pBidiC – Pointer to the Bi-di Object.

idRequest – Request ID.

0 – 65535	Reserved
65536 or above	Defined by each developer

pData – Pointer to the data for each request.

nDataBytes – Number of bytes of the data.

Corresponding API

Bi-di Plug-in Shared Library Interface

Bi-di Plug-in API	Shared Library Interface
bidiNew()	bidiLibNew()
bidiDestroy()	bidiLibDestroy()
bidiGetCap()	bidiLibGetCap()
bidiStartJob()	bidiLibStartJob()
bidiEndJob()	bidiLibEndJob()
bidiCancelJob()	bidiLibCancelJob()
bidiGetReadFD()	bidiLibGetReadFD()
bidiStartRead()	bidiLibStartRead()
bidiRead()	bidiLibRead()
bidiEndRead()	bidiLibEndRead()
bidiGetWriteFD()	bidiLibGetWriteFD()
bidiStartWrite()	bidiLibStartWrite()
bidiWrite()	bidiLibWrite()
bidiEndWrite()	bidiLibEndWrite()
bidiCtrl()	bidiLibCtrl()

Bi-di Plug-in Shared Library Interface

- Create a new extended object

Synopsis

```
void *bidiLibNew(int fdRead, int fdWrite, char *pURI);
```

Arguments

fdRead – File descriptor for reading the printer status data.

fdWrite – File descriptor for writing the printer command data.

pURI – Device URI.

Return Value

Void pointer to the extended object or NULL when error.

Bi-di Plug-in Shared Library Interface

- Other shared library functions

Arguments

First argument – A void pointer to the extended object created by the `bidiLibNew()` function.

Other arguments – Same arguments as the equivalent function of the Bi-di Plug-in API provides.

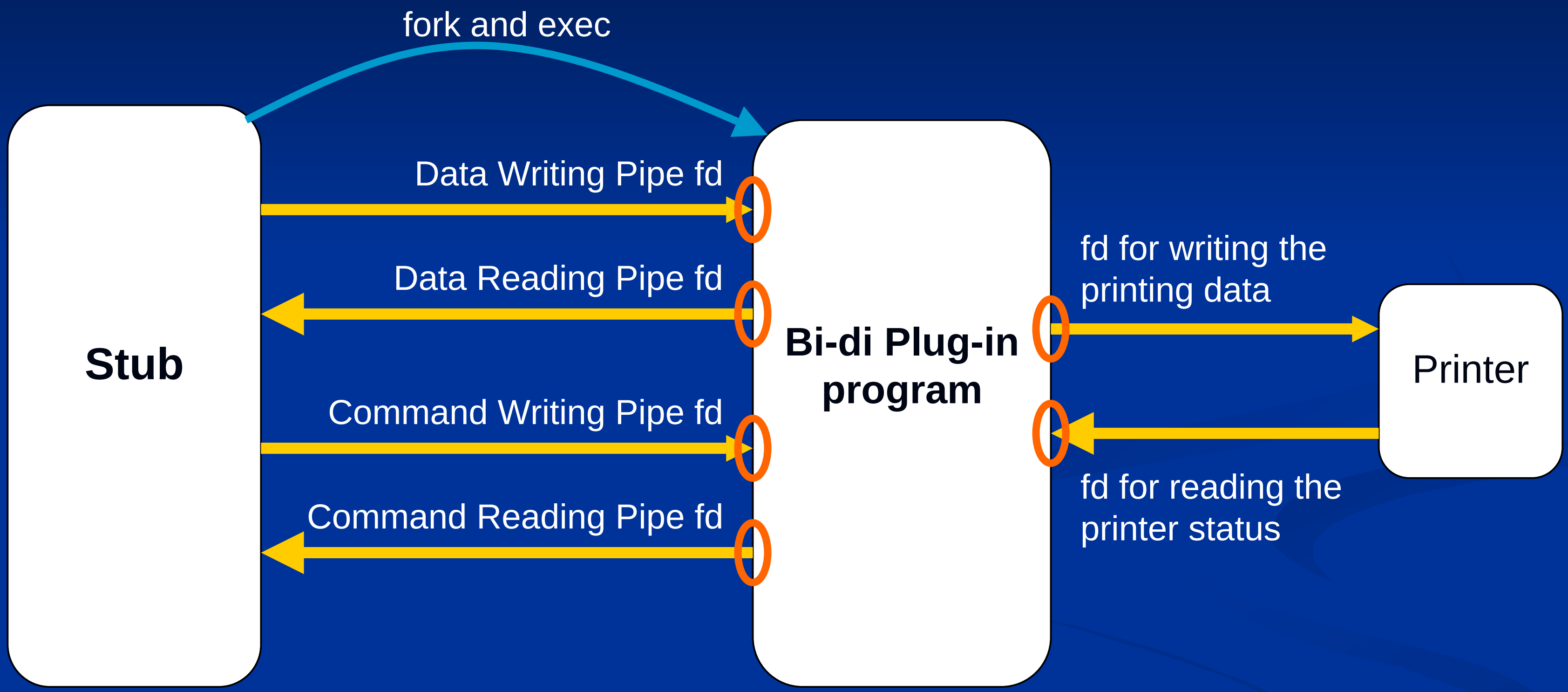
Functionality

Provides the same functionality as the equivalent function of the Bi-di Plug-in API provides.

Bi-di Plug-in Process Interface

- Stub forks the Bi-di Plug-in process when creating the Bi-di Object
- Stub pass the following arguments to the process
  - Data writing pipe file descriptor
  - Data reading pipe file descriptor
  - Command writing pipe file descriptor
  - Command reading pipe file descriptor
  - File descriptor for writing the printing data
  - File descriptor for reading the printer status
  - Device URI (Optional)

Bi-di Plug-in Process Interface



Bi-di Plug-in Process Interface

- Command reading/writing pipe are used for exchanging the command packet to execute the functions in the Bi-di Plug-in program.

- Each function is executed by one or more “Request” and “Acknowledge” command sequences.
  - Command packet format

Byte offset	Description
0 – 3	Command ID
4 – 7	Data length (n bytes)
8 – (8+n-1)	Data for each command

Bi-di Plug-in Process Interface

Bi-di Plug-in API	Process Interface Command ID
bidiNew()	BIDI_CMD_NEW
bidiDestroy()	BIDI_CMD_DESTROY
bidiGetCap()	BIDI_CMD_GETCAP
bidiStartJob()	BIDI_CMD_STARTJOB
bidiEndJob()	BIDI_CMD_ENDJOB
bidiCancelJob()	BIDI_CMD_CANCELJOB
bidiGetReadFD()	N/A
bidiStartRead()	BIDI_CMD_STARTREAD
bidiRead()	BIDI_CMD_READ
bidiEndRead()	BIDI_CMD_ENDREAD
bidiGetWriteFD()	N/A
bidiStartWrite()	BIDI_CMD_STARTWRITE
bidiWrite()	BIDI_CMD_WRITE
bidiEndWrite()	BIDI_CMD_ENDWRITE
bidiCtrl()	BIDI_CMD_CTRL

Printer Status Data Format

Printer Status Data Format

- Current version’s API supports:
  - The XML data according to the Printer MIB v2 Schema being considered in the PWG WBMM group. ftp://ftp.pwg.org/pub/pwg/wbmm/schemas/
  - The filtering mechanism to reduce a high data processing/transfer rate.
    - idReadMode of the `bidiStartRead()` function
      - BIDI_READ_PRT_MIB_ALL
      - BIDI_READ_PRT_MIB_SUMMARY

Issues/Concerns, Next and Info.

Issues / Concerns

- Need to describe:
  - The mapping rule of some of the printer status data to IPP attributes.
  - The PPD keywords related to Bi-di Plug-in modules.
  - Guidelines for saving the printer status data as a local file.
- Need to discuss:
  - The standard "idRequest" value for the `bidiCtrl()` function. (Currently, no standard idRequest is defined.)
  - Other printer status/information we need.
    - e.g.) UPDF
  - Coordination with other PWG/FSG standards.

Next

- Update the document according to the functionalities we need to append / modify.

Info.

- Contributors

TORATANI Yasumasa	Canon Inc.
Osamu MIHARA	FUJI XEROX Printing Systems
KANJO Hidenori	BBR INC.
YOSHIDA Mikio	BBR INC.
Shinpei KITAYAMA	EPSON KOWA
YAMAGISHI Toshihiro	Turbolinux
Hisao NAKAMURA	E&D
Koji OTANI	AXE

- Latest document

  ftp://ftp.pwg.org/pub/pwg/fsg/bidi/