

# Open Printing Bi-di Plug-in Application Program Interface (Bi-Di API) Specification

Draft Version 2004-03-01

## **Open Printing Bi-di Plug-in Application Interface Specification**

Copyright © 2004 Free Standards Group

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Portions of the text were taken from other copyrighted documents in accordance with the respective licenses of those documents.

Linux is a trademark of Linus Torvalds.

UNIX a registered trademark of the Open Group in the United States and other countries.

OpenGL is a registered trademark of Silicon Graphics, Inc.

PostScript is a registered trademark of Adobe Systems Inc.

# Table of Contents

<b>1.</b>	<b>NOTATION AND TERMINOLOGY</b> .....	<b>1</b>
1.1	NOTATION CONVENTIONS .....	1
1.2	CONFORMANCE TERMINOLOGY.....	1
1.3	OTHER TERMINOLOGY .....	1
<b>2.</b>	<b>INTRODUCTION</b> .....	<b>2</b>
2.1	SUMMARY.....	2
2.2	ARCHITECTURE OVERVIEW .....	2
<b>3.</b>	<b>BI-DI PLUG-IN API</b> .....	<b>3</b>
3.1	bidisNew .....	3
3.2	bidDestroy .....	3
3.3	bidGetCap .....	4
3.4	bidStartJob.....	4
3.5	bidEndJob .....	5
3.6	bidCancelJob.....	6
3.7	bidGetReadFD .....	6
3.8	bidStartRead.....	7
3.9	bidRead.....	8
3.10	bidEndRead .....	9
3.11	bidGetWriteFD .....	9
3.12	bidStartWrite.....	10
3.13	bidWrite .....	10
3.14	bidEndWrite .....	11
3.15	bidCtrl.....	12
<b>4.</b>	<b>BI-DI PLUG-IN SHARED LIBRARY INTERFACE</b> .....	<b>13</b>
4.1	Corresponding functions.....	13
4.2	bidLibNew.....	13
4.3	Other functions .....	13
<b>5.</b>	<b>BI-DI PLUG-IN PROCESS INTERFACE</b> .....	<b>14</b>
5.1	Pipe and Command Line Options .....	14
5.2	Command Format .....	15
5.3	BIDI_CMD_NEW.....	16
5.4	BIDI_CMD_DESTROY.....	17
5.5	BIDI_CMD_GETCAP .....	17
5.6	BIDI_CMD_STARTJOB .....	17
5.7	BIDI_CMD_ENDJOB .....	18
5.8	BIDI_CMD_CANCELJOB .....	18
5.9	BIDI_CMD_STARTREAD .....	19
5.10	BIDI_CMD_ENDREAD .....	19
5.11	BIDI_CMD_READ.....	19
5.12	BIDI_CMD_STARTWRITE.....	20
5.13	BIDI_CMD_ENDWRITE .....	20
5.14	BIDI_CMD_WRITE.....	21
5.15	BIDI_CMD_CTRL .....	21
5.16	Signal Handling.....	23
<b>6.</b>	<b>PRINTER STATUS DATA FORMAT</b> .....	<b>24</b>
6.1	Printer Status Data XML Schema .....	24
6.2	Printer Status Data Filtering.....	24
<b>7.</b>	<b>CONSTANTS</b> .....	<b>25</b>
7.1	Return Values.....	25
<b>8.</b>	<b>CONTRIBUTORS AND AUTHOR</b> .....	<b>26</b>
8.1	Contributors .....	26
8.2	Author.....	26

# 1. Notation and Terminology

## 1.1 Notation Conventions

This section describes the use of courier, italic and bold fonts in this document.

Font	Description	Examples
Courier	Definition of functions.	<code>void bidiDestroy(BidiC *pBidiC);</code>
	Name of functions.	<code>select()</code>
	Source code examples.	<code>#include &lt;opbidi.h&gt; BidiC *pBidiC = bidiNew("sample", 0, 1, NULL);</code>
Italic	Arguments of the function used in the description.	<i>nBufBytes</i>
	File, directory or other resource names.	<i>/var/log/opbidi/printer</i>
Bold	Emphasis	

## 1.2 Conformance Terminology

In this document, capitalized terms, such as: MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, MAY, and OPTIONAL, are intended to be interpreted as described in [RFC2119].

## 1.3 Other Terminology

The following table defines special terms used in this document.

Term	Description
Bi-di Plug-in module	A shared library or an individual program which provides a set of the Bi-di Plug-in API functions.
Bi-di Plug-in library	A Bi-di Plug-in module as a shared library being linked by the Stub.
Bi-di Plug-in program	A Bi-di Plug-in module as an individual program which communicates with the Stub via an inter-process communication interface.
Stub	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.
Monitor	A program which calls the Bi-di Plug-in API entries provided by the Stub.
Bi-di Object	An object which keeps a data set created by the Stub for controlling the Bi-di Plug-in module.
Bi-di Plug-in process	A process of the Bi-di Plug-in program.

## 2. Introduction

### 2.1 Summary

This document defines the API for obtaining the printer status data from the printer in the standard data format, and the API for writing the printer command data to the printer as well. This document also defines the standard data format to describe the printer status data.

### 2.2 Architecture Overview

The following figure shows the modules related to the Bi-di Plug-in API.

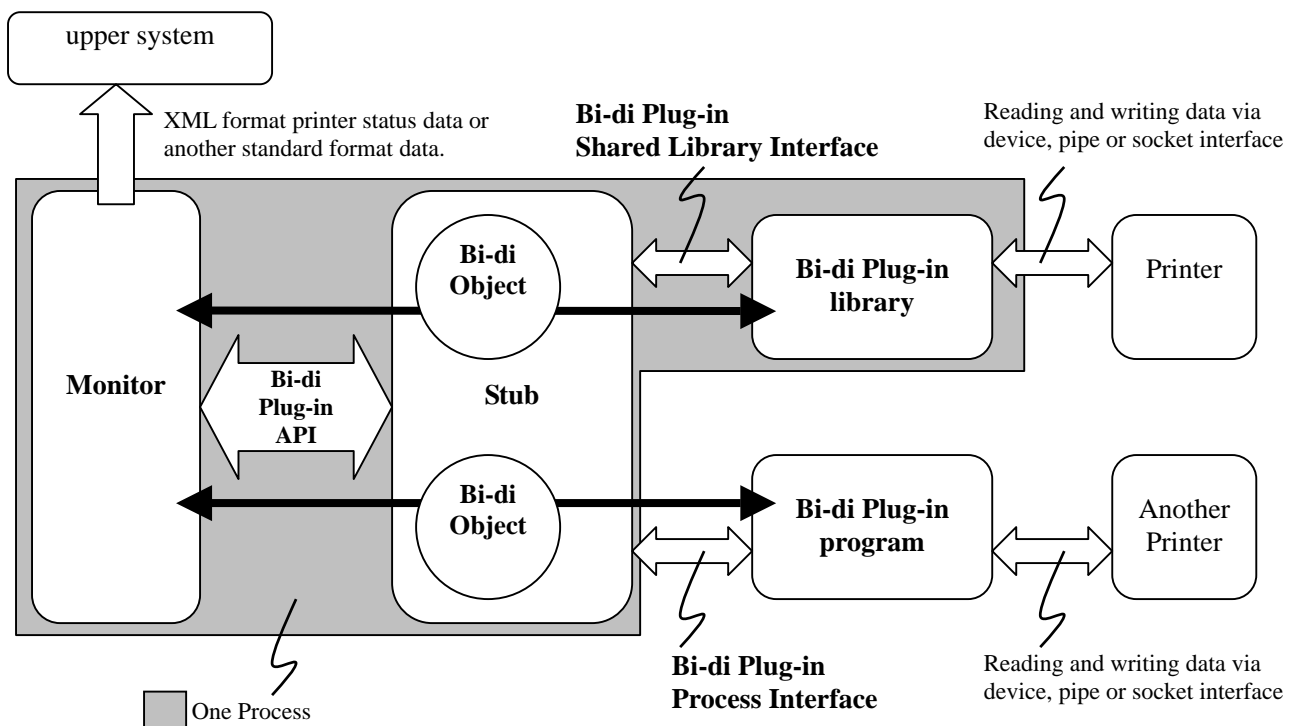


Fig.1 Architecture Overview

The 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. To obtain the printer status from the printer, the Monitor creates the Bi-di Object with the Bi-di Plug-in API. The Bi-di Object is an object which keeps a data set for controlling the Bi-di Plug-in module. The Monitor can create multiple Bi-di Objects, however one Bi-di Object can deal with only one Bi-di Plug-in module. The Bi-di Plug-in library is a Bi-di Plug-in module provided as a shared library. The Stub loads and links the Bi-di Plug-in library when creating the Bi-di Object, and communicates with it via the Bi-di Plug-in Shared Library Interface. Therefore, the Monitor, the Stub and the Bi-di Plug-in library run as the same process. Another type of Bi-di Plug-in module is the Bi-di Plug-in Program that is an individual program loaded and executed by the Stub when creating the Bi-di Object. The Bi-di Plug-in program communicates with the Stub via the Bi-di Plug-in Process Interface.

The Monitor obtains the printer status data from the Bi-di Plug-in module at an appropriate interval time. Raw printer status data just obtained from the printer via a device, pipe or socket interface depends on each printer model or protocol. The Bi-di Plug-in module converts the raw printer status data into the device independent XML format status data, and the Monitor can obtain it via the Bi-di Plug-in API. Furthermore, the Monitor can send the XML format printer status data to an upper system, such as an application program or a spooler program, and the upper system can parse and use the data. In other cases, the Monitor can parse the XML format printer status data by itself and convert it into another standard format data, such as IPP, and send it to the upper system, so that the upper system can deal with the device independent printer status data.

The following section in this document describes the Bi-di Plug-in API, Bi-di Plug-in Shared Library Interface, Bi-di Plug-in Process Interface and the Printer Status Data Format in detail.

## 3. Bi-di Plug-in API

### 3.1 bidiNew

#### Name

bidiNew – Create a new Bi-di Object. <This function 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.

#### Description

- This function creates the Bi-di Object specified by *pName* and return the printer to it.  
pName で指定される Bi-di object を生成してそのポインタを返す。
- First, this function MUST assume *pName* as the name of the Bi-di Plug-in library. The appropriate prefix (e.g. lib\*) and postfix (e.g. .so) for the library name for each operating system are automatically appended in this function, and this function tries to load and link the Bi-di Plug-in library from the appropriate library path under each operating system.  
本関数では、先ず pName に library type の Bi-di Plug-in module が指定されたとみなして処理する。動作環境におけるライブラリ名の先頭の定型文字列(lib など)および末尾の定型文字列(.so)は、本関数の内部処理において自動的に付加し、動作環境での適切な library path からの Bi-di Plug-in library module の load と link を試みる。
- If this function fails to load or link to the Bi-di Plug-in library, this function MUST assume *pName* as the name of the Bi-di Plug-in program, and this function tries to load and execute the Bi-di Plug-in program from the appropriate executable path under each operating system.  
library type の Bi-di Plug-in module の load または link に失敗した場合は、次に pName に process type の Bi-di Plug-in module が指定されたとみなして処理する。process type の場合は pName に与えられた文字列をそのまま Bi-di Plug-in module のファイル名としてみなし、動作環境での適切な実行 path からの Bi-di Plug-in module の実行を試みる。
- This function MAY be blocked until all the initialization procedures of the Bi-di Plug-in module are completed, or return a NULL pointer immediately without the initialization of it after receiving the signal.  
一連の処理が完了するまで本関数はブロックするが、シグナルの受信によって処理未完了のまま本関数から戻る場合がある。その場合、本関数の戻り値は NULL となる。
- This function MAY be blocked depending on the connection process between the Bi-di Plug-in module and the printer. The Monitor SHOULD set the alarm signal or other time out mechanism before calling this function.  
Bi-di Plug-in module がプリンタとのコネクションを確立する処理によっては、本関数は長時間ブロックする可能性がある。呼び元が一定時間内の応答を規定したい場合は、本関数を呼ぶ前にアラームを設定する等する。
- The writing functionality of the Bi-di Plug-in module is optional. The Monitor can know whether the Bi-di Plug-in module supports the writing functionality or not by the `bidiGetCap()` function. When the writing functionality is not supported, the Monitor MUST write the printer command data directly to the printer via an appropriate file descriptor.  
Bi-di Plug-in の write function はオプション。Bi-di Plug-in 側が write function をサポートするか否かは `bidiGetCap` 関数によって呼び元が判別する。write function がサポートされない場合は呼び元が適切な file descriptor を用いて直接 printer command data を write する必要がある。
- *pURI* is OPTIONAL. NULL for *pURI* means that the device URI is unknown and the way of how to deal with the device URI depends on the implementation of each Bi-di Plug-in module.  
pURI で指定するデバイスの URI はオプション。URI が不明の場合等は NULL の指定を許す。NULL が指定された場合の Bi-di Plug-in 側の処理は実装依存とする。

#### Return Value

Pointer to the Bi-di Object or NULL when error.

### 3.2 bidiDestroy

#### Name

bidiDestroy – Destroy the Bi-di Object. <This function MUST be supported.>

#### Synopsis

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

#### Arguments

pBidiC – Pointer to the Bi-di Object.

### Description

- When the Bi-di Object specified by *pBidiC* is the Bi-di Plug-in program, this function kills it and destroys the Bi-di Object.
- When the Bi-di Object specified by *pBidiC* is the Bi-di Plug-in library, this function unlinks it and destroys the Bi-di Object. *pBidiC* によって指定した Bi-di Object によって指定される Bi-di Plug-in module が process type の場合はそれを kill し、library type の場合はそれを unlink した後、Bi-di object を解放する。
- This function MAY be blocked depending on the disconnection process between the Bi-di Plug-in module and the printer. The Monitor SHOULD set the alarm signal or other time out mechanism before calling this function. Bi-di Plug-in module がプリンタとのコネクションを解放する処理によっては、本関数は長時間ブロックする可能性がある。呼び元が一定時間内の応答を規定したい場合は、本関数を呼ぶ前にアラームを設定する等する。

### Return Value

None

## 3.3 bidiGetCap

### Name

bidiGetCap – Get the Bi-di Object capabilities. <This function MUST be supported.>

### Synopsis

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

### Arguments

*pBidiC* – Pointer to the Bi-di Object.

*cap* – Enum of the Bi-di Object capabilities. This function MUST support the following enums.

cap	Value	Description
BIDI_CAP_WRITE	1	Supports the <code>bidiStartWrite()</code> , <code>bidiEndWrite()</code> and the <code>bidiWrite()</code> functions.
BIDI_CAP_JOB	2	Supports the <code>bidiStartJob()</code> , <code>bidiEndJob()</code> and the <code>bidiCancelJob()</code> functions.
BIDI_CAP_CTRL	3	Supports the <code>bidiCtrl()</code> function.

### Description

- This function returns the value which indicates whether the condition specified by *cap* is true or false. *pBidiC* によって指定した Bi-di Object によって指定される Bi-di Plug-in module において、*cap* で指定した条件が正しいか否かを返す。
- The Bi-di Plug-in program MUST support the `bidiStartWrite()`, `bidiEndWrite()` and the `bidiWrite()` functions. This function MUST return BIDI\_TRUE when the Bi-di Object specified by *pBidiC* is the one for the Bi-di Plug-in program. process type の Bi-di Plug-in module の場合、`bidiStartWrite`, `bidiEndWrite`, `bidiWrite` 関数のサポートは必須とし、*cap* に BIDI\_CAP\_WRITE が指定されて本関数が呼ばれた場合は、必ず BIDI\_TRUE を返さなければならないものとする。
- This function MUST NOT be blocked. 本関数はブロックしない。

### Return Value

BIDI\_TRUE      The condition specified by *cap* is true.  
cap で指定した条件が正

BIDI\_FALSE     The condition specified by *cap* is false.  
cap で指定した条件が誤

BIDI\_ERROR     Any error.  
エラー

The Monitor MUST destroy the Bi-di Object when this function returns BIDI\_ERROR.

BIDI\_ERROR 発生時は、呼び元は `bidiDestroy` 関数によって Bi-di Object を破棄しなければならない。

## 3.4 bidiStartJob

### Name

bidiStartJob – Start the printing job sequence. <This function is OPTIONAL.>

### Synopsis

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

### Arguments

*pBidiC* – Pointer to the Bi-di Object.

idJob – Job ID.

### **Description**

- The Monitor declares the start of the printing job to the Bi-di Plug-in module specified by *pBidiC*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module に対してジョブの開始を通知する。
- This function MUST be called before starting printing job by this function and after the end of printing job by calling the `bidEndJob()` function, or return `BIDI_ERROR`.  
本関数が呼ばれた後、`bidEngJob` 関数によってジョブを終了する前に再び `bidStartJob` 関数が呼ばれた場合はエラーとなり、戻り値として `BIDI_ERROR` を返す。
- This function MUST NOT be blocked.  
本関数はブロックしない。

### **Return Value**

<code>BIDI_OK</code>	Success. 正常終了
<code>BIDI_EPROGRESS</code>	The printing job specified <i>idJob</i> is in progress and cannot start a new job with the same job ID. 指定したジョブが処理中
<code>BIDI_ERROR</code>	Other errors. その他のエラー

The Monitor MUST destroy the Bi-di Object when this function returns `BIDI_ERROR`.

`BIDI_ERROR` 発生時は、呼び元は `bidDestroy` 関数によって Bi-di Object を破棄しなければならない。

## **3.5 bidEndJob**

### **Name**

`bidEndJob` – Stop the printing job sequence. <This function is OPTIONAL. If the `bidStartJob()` function is supported, this function MUST be supported.>

### **Synopsis**

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

### **Arguments**

`pBidiC` – Pointer to the Bi-di Object.

### **Description**

- The Monitor checks the end of the printing job to the Bi-di Plug-in module specified by *pBidiC* by this function.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module に対してジョブの終了を通知する。
- The Monitor MUST call this function after finishing or canceling a printing job.  
本関数は、`bidStartJob` 関数が呼ばれた後の一連のジョブの処理完了後、必ず呼ばなければならない。
- This function returns `BIDI_ERROR` if the `bidStartJob()` function is not called by the Monitor or a printing job is not executed prior to calling this function.  
`bidStartJob` 関数が呼ばれていない状態で本関数が呼ばれた場合はエラーとなり、戻り値として `BIDI_ERROR` を返す。
- This function MUST NOT be blocked.  
本関数はブロックしない。

### **Return Value**

<code>BIDI_OK</code>	Success. 正常終了
<code>BIDI_EPROGRESS</code>	The printing job specified <i>idJob</i> is in progress. ジョブが処理中で終了できない
<code>BIDI_ERROR</code>	Other errors. その他のエラー

The Monitor MUST destroy the Bi-di Object when this function returns `BIDI_ERROR`.

`BIDI_ERROR` 発生時は、呼び元は `bidDestroy` 関数によって Bi-di Object を破棄しなければならない。

Whether this function returns `BIDI_EPROGRESS` or not depends on the specification of each Bi-di Plug-in module. When this function returns `BIDI_EPROGRESS`, the Monitor MUST call this function again to try to check the end of the printing job after the specified interval time. The specified interval time depends on the implementation. The Bi-di Plug-in module MUST provide all the same functionalities, including the reading printer status data by the `bidRead()` function, that the Bi-di Plug-in module provides before `BIDI_EPROGRESS` occurs.

`BIDI_EPROGRESS` が発生するか否かは、Bi-di Plug-in module の仕様に依存する。尚、本エラー発生時は、呼び元は一定時間経過後、再度本関数を呼んでジョブの終了を試みなければならない。また Bi-di Plug-in module が本エラーを発生する場合は、その発生後も `bidRead` 関数によるプリンタステータスデータの取得等の全ての処理を、本エラー発生前と変わりなく継続して提供しなければならない。



## 3.6 bidiCancelJob

### Name

bidiCancelJob – Cancel the printing job sequence. <This function is OPTIONAL. If the bidiStartJob() function is supported, this function MUST be supported.>

### Synopsis

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

### Arguments

pBidiC – Pointer to the Bi-di Object.  
idJob – Job ID.

### Description

- The Monitor declares the cancellation of the printing job specified by *idJob* to the Bi-di Plug-in module specified by *pBidiC*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module に対して、idJob で指定したジョブの中断を通知する。
- This function MUST NOT be blocked.  
本関数はブロックしない。

### Return Value

BIDI_OK	Success. 正常終了
BIDI_ENOJOB	The printing job specified by the <i>idJob</i> is no longer in progress. Bi-di Plug-in module にジョブが存在しない
BIDI_EPROGRESS	The printing job specified by the <i>idJob</i> is in progress and cannot cancel the job. ジョブが中断できない
BIDI_ERROR	Other errors. その他のエラー

The Monitor MUST destroy the Bi-di Object when this function returns BIDI\_ERROR.

BIDI\_ERROR 発生時は、呼び元は bidiDestroy 関数によって Bi-di Object を破棄しなければならない。

Whether this function returns BIDI\_ENOJOB or not depends on the specification of each Bi-di Plug-in module. If the Bi-di Plug-in module has the capability of receiving the multiple printing jobs, the Bi-di Plug-in module SHOULD retain each job ID for each printing job, and cancel only the printing job specified by *idJob* when the Monitor calls this function, or return BIDI\_ENOJOB when the job is no longer in progress or *idJob* is a nonexistent job ID. If the Bi-di Plug-in module does not have the capability of receiving the multiple printing jobs, the Bi-di Plug-in module retains only one job ID for one printing job at the same time, and cancel the printing job specified by *idJob* when the Monitor calls this function, or return BIDI\_ENOJOB when the job is no longer in progress or *idJob* is a nonexistent job ID. The way of how to deal with BIDI\_ENOJOB depends on the Monitor. BIDI\_ENOJOB が発生するか否かは、Bi-di Plug-in module の仕様に依存する。複数のジョブをスプール可能な Bi-di Plug-in module の場合、Bi-di Plug-in module はスプールおよび印刷中のジョブの数分だけ Job ID を保持し、それらのうちの1つが指定された場合は対応するジョブを中断し、それら以外の Job ID が指定された場合は、本エラーを返さなければならない。また、単一のジョブのみ処理が可能な Bi-di Plug-in module は単一の Job ID を保持し、その Job ID が指定された場合は処理中のジョブを中断し、それ以外の Job ID が指定された場合は本エラーを返さなければならない。尚、本エラー発生時の呼び元の処理は、実装依存とする。

This function returns BIDI\_EPROGRESS when the printing job specified by *idJob* is in progress and the Bi-di Plug-in module cannot cancel the job. The way of how to deal with BIDI\_EPROGRESS depends on the Monitor.

BIDI\_EPROGRESS は、Bi-di Plug-in module 側に idJob に対応するジョブが存在するにも関わらず、ジョブが中断できなかった場合に発生する。本エラー発生時の呼び元の処理は、実装依存とする。

## 3.7 bidiGetReadFD

### Name

bidiGetReadFD – Get the file descriptor for reading from the Bi-di Object. <This function MUST be supported.>

### Synopsis

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

### Arguments

pBidiC – Pointer to the Bi-di Object.

### Description

- This function returns the file descriptor for reading the printer status data from the Bi-di Object specified by *pBidiC*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module の read 用 file descriptor を返す。

- The file descriptor obtained from this function is valid until the Monitor destroys the Bi-di Object.  
取得した file descriptor は bidiDestroy 関数によって Bi-di Object を破棄するまでの間有効。
- The Monitor can use the file descriptor obtained from this function for the UNIX select ( ) function to determine its own processing.  
呼び元は本関数によって取得した file descriptor を用いて select 関数による read 処理への振り分けを行うことができる。
- This function MUST NOT be blocked.  
本関数はブロックしない。

### Return Value

File descriptor.  
BIDI\_ERROR Any error.  
エラー

The Monitor MUST destroy the Bi-di Object when this function returns BIDI\_ERROR.  
BIDI\_ERROR 発生時は、呼び元は bidiDestroy 関数によって Bi-di Object を破棄しなければならない。

## 3.8 bidiStartRead

### Name

bidiStartRead – Start the reading sequence. <This function 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.  
pLang – Pointer to the language string.

### Description

- The Monitor declares the start of reading the printer status data to the Bi-di Plug-in module specified by *pBidiC*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module からの、プリンタステータスデータの read の開始を Bi-di Plug-in module に通知する。
- *idReadMode* is the enum of the reading mode. This function SHOULD support the following enums.

idReadMode	Value	Description
BIDI_READ_PRT_MIB_ALL	255	Read all the printer status data.
BIDI_READ_PRT_MIB_SUMMARY	1	Read the summarized printer status data.

- The Monitor MUST set the “language”, “region” and “character code” string, (e.g. “ja\_JP.UTF-8”) defined by the locale specification to *pLang*. The Bi-di Plug-in module MUST generate the printer status data according to the specified language, region and character code. The Bi-di Plug-in module MUST support UTF-8 at least. The way of how to deal with the other character codes, (e.g. ja\_JP.eucJP) depends on each Bi-di Plug-in module implementation.  
pLang には、”ja\_JP.UTF-8”のようなロケール指定文字列と同様の言語 + 地域 + 文字コード指定文字列を指定する。Bi-di Plug-in module は、pLang によって指定された言語および地域情報に従ってプリンタステータスデータを生成する。また、Bi-di Plug-in module は、少なくとも UTF-8 の文字コードに従ったステータスデータを生成しなければならない。pLang に、”ja\_JP.eucJP”のような UTF-8 とは異なる文字コード指定を含んだ文字列が与えられた場合、Bi-di Plug-in module が指定された文字コードでステータスデータを生成するか否かは実装依存とする。
- The Bi-di Plug-in module MUST generate the printer status data in English when the Monitor set a NULL value to *pLang* or when the Bi-di Plug-in module cannot generate the printer status data according to the language and region specified by *pLang*.  
pLang に NULL が指定された場合や、pLang によって指定された言語および地域に従ったプリンタステータスデータの生成が不可能な場合は、Bi-di Plug-in module は少なくとも英語によるプリンタステータスデータを生成しなければならない。
- The Bi-di Plug-in module MUST establish and retain the printer status data to be read by the Monitor via the *bidiRead ( )* function.  
Bi-di Plug-in module は本関数が呼ばれたタイミングで、呼び元に返すプリンタステータスデータを確定する。
- When the Bi-di Plug-in module is a Bi-di Plug-in program, the Bi-di Plug-in program MUST write 0(one byte) into the Data Reading Pipe after the Bi-di Plug-in program establishes and retains the printer status data, and this function MUST first read the above one byte from the Data Reading Pipe specified by the file descriptor which can be obtained by the *bidiGetReadFD ( )* function.  
process type の Bi-di Plug-in の場合、Bi-di Plug-in module はプリンタステータス送用の準備が出来た時点で、それを表す 1 バイトの値 0 を呼び元の read 用の file descriptor に繋がるパイプに書き込まなければならない。呼び元にリンクする *bidiStartRead* 関数の内部処理では、*bidiGetReadFD* 関数によって取得できる file descriptor から上記 1 バイトの値 0 を読み捨てなければならない。
- This function MUST return BIDI\_ERROR when the Monitor calls the following functions before the Monitor calls the

`bidiEndRead()` function.

`bidiStartRead()`, `bidiStartWrite()`, `bidiEndWrite()`

本関数は必ず `bidiEndRead` 関数と対で呼ばれなければならない。本関数が呼ばれてから `bidiEndRead` 関数が呼ばれるまでの間に次の関数が呼ばれた場合はエラーとなり、戻り値として `BIDI_ERROR` を返す。

`bidiStartRead`, `bidiStartWrite`, `bidiEndWrite`

- This function MAY be blocked until the Bi-di Plug-in module establishes and retains the printer status data, or return a `BIDI_INTR` error caused by receiving the signals.  
プリンタステータスデータが確定するまで本関数はブロックするが、シグナルの受信によって処理の途中で本関数から戻る場合には、戻り値として `BIDI_INTR` を返す。
- This function MAY be blocked depending on the connection process between the Bi-di Plug-in module and the printer. The Monitor SHOULD set the alarm signal or other time out mechanism before calling this function.  
`Bi-di Plug-in module` がプリンタとのコネクションを確立する処理によっては、本関数は長時間ブロックする可能性がある。呼び元が一定時間内の応答を規定したい場合は、本関数を呼ぶ前にアラームを設定する等する。

### Return Value

<code>BIDI_OK</code>	Success. 正常終了
<code>BIDI_INTR</code>	Receive a signal. シグナル受信
<code>BIDI_ERROR</code>	Other errors. その他のエラー

The Monitor MUST destroy the Bi-di Object when this function returns `BIDI_ERROR`.

`BIDI_ERROR` 発生時は、呼び元は `bidiDestroy` 関数によって Bi-di Object を破棄しなければならない。

## 3.9 bidiRead

### Name

`bidiRead` – Read the printer status data from the Bi-di Object. <This function 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.

### Description

- This function obtains the printer status data from the Bi-di Object specified by `pBidiC` and store the data into the buffer specified by `pBuf`. The data format of the printer status data is defined in the “Printer Status Data Format” section.
- `pBidiC` によって指定した Bi-di Object によって指定される Bi-di Plug-in module から、プリンタステータスデータを取得して、`pBuf` で指定したバッファにデータを格納する。プリンタステータスデータのデータフォーマットについては、Printer Status Data Format のセクションを参照。
- When the length of the printer status data is longer than the length of the buffer specified by `nBufBytes`, this function read the `nBufBytes` bytes data from the top of the printer status data first and store it at the top of the buffer, and next time this function is called, this function reads the remaining printer status data from the next byte of the previous read data and stores it in the next address of the end of the previous written data into the buffer.
- 指定したバッファのバイト数を越えるプリンタステータスデータがある場合には、格納できるバイト数だけ `pBuf` の先頭から格納し、次に再度本関数が呼ばれた際には、その次のバイト位置のプリンタステータスデータから、格納できるバイト数だけ `pBuf` の先頭から格納する。
- When the remaining printer status data is shorter than the length specified by `nBufBytes`, this function returns the specified shorter byte number instead of the `nBufBytes`, or returns zero when no printer status data remains.  
`nBufBytes` で指定した値よりもプリンタステータスデータのバイト数が小さいような場合は、`nBufBytes` で指定した値よりも小さい値を返す場合がある。その後更に本関数を呼ぶと、戻り値として 0 を返す。
- When receiving the signals, this function MAY read and store the printer status data which is shorter than the length specified by `nBufBytes`, and return the shorter length of the data that this function actually read and stored.  
プリンタステータスデータの read が完了するまで本関数はブロックするが、シグナルの受信によってプリンタステータスデータの read の途中で本関数から戻る場合には、`nBufBytes` で指定した値よりも小さい値を返す場合がある。
- This function returns `BIDI_INTR` when no printer status data is read regardless of the printer status data that should be read retains.  
1 バイトも read せずにシグナルを受信した場合は、戻り値として `BIDI_INTR` を返す。
- This function MAY be blocked depending on the connection process between the Bi-di Plug-in module and the printer. The Monitor SHOULD set the alarm signal or other time out mechanism before calling this function.

Bi-di Plug-in module がプリンタとのコネクションを確立する処理によっては、本関数は長時間ブロックする可能性がある。呼び元が一定時間内の応答を規定したい場合は、本関数を呼ぶ前にアラームを設定する等する。

- The Monitor MUST call this function between the `bidistartRead()` and `bidiread()` functions. This function MUST return `BIDI_ERROR` when this function is not called between the `bidistartRead()` and `bidiread()` functions.

本関数は `bidistartRead` 関数と `bidiread` 関数の間で呼ばなければならない。その範囲外で呼ばれた場合は、戻り値として `BIDI_ERROR` を返す。

### **Return Value**

Number of bytes of the printer status data read, or 0 when no data remains.

`BIDI_EINTR` Receive a signal with no data read.

0バイト read でのシグナル受信

`BIDI_ERROR` Other errors.

その他のエラー

## **3.10 bidiEndRead**

### **Name**

`bidiEndRead` – Stop the reading sequence. <This function MUST be supported.>

### **Synopsis**

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

### **Arguments**

`pBidiC` – Pointer to the Bi-di Object.

### **Description**

- The Monitor declares the end of reading the printer status data to the Bi-di Plug-in module specified by `pBidiC`.  
`pBidiC` によって指定した Bi-di Object によって指定される Bi-di Plug-in module からの、プリンタステータスデータの read の完了を Bi-di Plug-in module に通知する。
- The Bi-di Plug-in module can update the printer status data after this function is called by the Monitor.  
Bi-di Plug-in module は本関数が呼ばれた後、呼び元に返すプリンタステータスデータの更新を再開して良い。
- The Monitor MUST call this function after calling the `bidistartRead()` function.  
本関数は、`bidistartRead` 関数が呼ばれた後の一連の read 処理完了後、必ず呼ばなければならない。
- This function returns `BIDI_ERROR` if the `bidistartRead()` function is not called by the Monitor prior to calling this function.  
`bidistartRead` 関数が呼ばれていない状態で本関数が呼ばれた場合はエラーとなり、戻り値として `BIDI_ERROR` を返す。
- This function MUST NOT be blocked.  
本関数はブロックしない。

### **Return Value**

`BIDI_OK` Success.

正常終了

`BIDI_ERROR` Any error.

エラー

The Monitor MUST destroy the Bi-di Object when this function returns `BIDI_ERROR`.

`BIDI_ERROR` 発生時は、呼び元は `bidiDestroy` 関数によって Bi-di Object を破棄しなければならない。

## **3.11 bidiGetWriteFD**

### **Name**

`bidiGetWriteFD` – Get the file descriptor for writing to the Bi-di Object. <This function MUST be supported.>

### **Synopsis**

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

### **Arguments**

`pBidiC` – Pointer to the Bi-di Object.

### **Description**

- This function returns the file descriptor for writing the printer command data to the Bi-di Object specified by `pBidiC`.  
`pBidiC` によって指定した Bi-di Object によって指定される Bi-di Plug-in module の write 用 file descriptor を返す。
- The file descriptor obtained from this function is valid until the Monitor destroys the Bi-di Object.  
取得した file descriptor は `bidiDestroy` 関数によって Bi-di Object を破棄するまでの間有効。

- The Monitor can use the file descriptor obtained from this function for the UNIX `select()` function to determine its own processing.  
呼び元は本関数によって取得した file descriptor を用いて select 関数による write 処理への振り分けを行うことができる。
- This function **MUST NOT** be blocked.  
本関数はブロックしない。

### **Return Value**

File descriptor.  
BIDI\_ERROR Any error.  
エラー

The Monitor **MUST** destroy the Bi-di Object when this function returns BIDI\_ERROR.  
BIDI\_ERROR 発生時は、呼び元は `bidDestroy` 関数によって Bi-di Object を破棄しなければならない。

## **3.12 bidiStartWrite**

### **Name**

`bidiStartWrite` – Start the writing sequence. <This function is OPTIONAL. If the `bidiWrite()` function is supported, this function **MUST** be supported.>

### **Synopsis**

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

### **Arguments**

`pBidiC` – Pointer to the Bi-di Object.

### **Description**

- The Monitor declares the start of writing the printer command data to the Bi-di Plug-in module specified by `pBidiC`.  
`pBidiC` によって指定した Bi-di Object によって指定される Bi-di Plug-in module への、データの write の開始を Bi-di Plug-in module に通知する。
- This function **MUST** return BIDI\_ERROR when the Monitor calls the following functions before the Monitor calls the `bidiEndWrite()` function.  
`bidiStartRead()`, `bidiStartWrite()`, `bidiEndRead()`  
本関数は必ず `bidiEndWrite` 関数と対で呼ばれなければならない。本関数が呼ばれてから `bidiEndWrite` 関数が呼ばれるまでの間に次の関数が呼ばれた場合はエラーとなり、戻り値として BIDI\_ERROR を返す。  
`bidiStartRead`, `bidiStartWrite`, `bidiEndRead`
- This function **MAY** be blocked depending on the connection process between the Bi-di Plug-in module and the printer. The Monitor **SHOULD** set the alarm signal or other time out mechanism before calling this function. This function returns BIDI\_EINTR when receiving the signals.
- Bi-di Plug-in module がプリンタとのコネクションを確立する処理によっては、本関数は長時間ブロックする可能性がある。呼び元が一定時間内の応答を規定したい場合は、本関数を呼ぶ前にアラームを設定する等する。シグナルの受信によって処理の途中で本関数から戻るときは、戻り値として BIDI\_EINTR を返す。

### **Return Value**

BIDI\_OK Success.  
正常終了  
BIDI\_EINTR Receive a signal.  
シグナル受信  
BIDI\_ERROR Other errors.  
その他のエラー

The Monitor **MUST** destroy the Bi-di Object when this function returns BIDI\_ERROR.  
BIDI\_ERROR 発生時は、呼び元は `bidDestroy` 関数によって Bi-di Object を破棄しなければならない。

## **3.13 bidiWrite**

### **Name**

`bidiWrite` – Write data to the Bi-di Object. <This function is OPTIONAL.>

### **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.

### Description

- This function reads the printer command data from the buffer specified by *pBuf* and write it into the Bi-di Object specified by *pBidiC*. The length of the reading and writing printer command data is specified by *nBufBytes*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module へ、pBuf で指定したバッファ中のデータを nBufBytes で指定したバイト数だけ書き込む。
- When receiving the signals, this function MAY read and write the printer command data which is shorter than the length specified by *nBufBytes*, and return the shorter length of the data that this function actually read and wrote. The Monitor MUST call this function again to write the rest of the printer command data into the Bi-di Object with the appropriate address in the buffer and the appropriate length of the printer command data.  
データの write が完了するまで本関数はブロックするが、シグナルの受信によってデータの write の途中で本関数から戻る場合には、nBufBytes で指定した値よりも小さい値を返す場合がある。その際、残りのデータを書き込むためには、呼び元がバッファの適切なバイト位置から、本関数によって再度データを書き込む必要がある。
- This function returns BIDI\_EINTR when no printer command data is written regardless of the printer command data that should be written retains.  
1 バイトも write せずにシグナルを受信した場合は、戻り値として BIDI\_EINTR を返す。
- This function MAY be blocked depending on the connection process between the Bi-di Plug-in module and the printer. The Monitor SHOULD set the alarm signal or other time out mechanism before calling this function.  
Bi-di Plug-in module がプリンタとのコネクションを確立する処理によっては、本関数は長時間ブロックする可能性がある。呼び元が一定時間内の応答を規定したい場合は、本関数を呼ぶ前にアラームを設定する等する。
- The Monitor MUST call this function between the *bidistartwrite()* and *bidendwrite()* functions. This function MUST return BIDI\_ERROR when this function is not called between the *bidistartwrite()* and *bidendwrite()* functions.  
本関数は *bidistartwrite* 関数と *bidendwrite* 関数の間で呼ばなければならない。その範囲外で呼ばれた場合は、戻り値として BIDI\_ERROR を返す。

### Return Value

Number of bytes of the writing data written or 0 when no data written.

BIDI_EINTR	Receive a signal with no data written. 0 バイト write でのシグナル受信
BIDI_ERROR	Other errors. その他のエラー

## 3.14 bidendwrite

### Name

*bidendwrite* – Stop the writing sequence. <This function is OPTIONAL. If the *bidewrite()* function is supported, this function MUST be supported.>

### Synopsis

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

### Arguments

*pBidiC* – Pointer to the Bi-di Object.

### Description

- The Monitor declares the end of writing the printer command data to the Bi-di Plug-in module specified by *pBidiC*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module への、データの write の完了を Bi-di Plug-in module に通知する。
- The Monitor MUST call this function after calling the *bidistartwrite()* function.  
本関数は、*bidistartwrite* 関数が呼ばれた後の一連の write 処理完了後、必ず呼ばなければならない。
- This function returns BIDI\_ERROR if the *bidistartwrite()* function is not called by the Monitor prior to calling this function.  
*bidistartwrite* 関数が呼ばれていない状態で本関数が呼ばれた場合はエラーとなり、戻り値として BIDI\_ERROR を返す。
- This function MUST NOT be blocked.  
本関数はブロックしない。

### Return Value

BIDI_OK	Success. 正常終了
BIDI_ERROR	Any error. エラー

The Monitor MUST destroy the Bi-di Object when this function returns BIDI\_ERROR.

BIDI\_ERROR 発生時は、呼び元は bidiDestroy 関数によって Bi-di Object を破棄しなければならない。

### 3.15 bidiCtrl

#### **Name**

bidiCtrl – Request the extended operation. <This function is OPTIONAL.>

#### **Synopsis**

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

#### **Arguments**

pBidiC – Pointer to the Bi-di Object.  
idRequest – Request ID.  
pData – Pointer to the data for each request.  
nDataBytes – Number of bytes of the data.

#### **Description**

- The Monitor requests the Bi-di Object specified by *pBidiC* to do the extended operation specified by *idRequest* and *nDataBytes*.  
pBidiC によって指定した Bi-di Object によって指定される Bi-di Plug-in module に対して、拡張制御を要求する。
- idRequest is defined as below.

idRequest	Category
0 ~ 65535	Reserved.
65536 or above	Defined by each developer.

#### **Return Value**

Number of bytes of the data read or 0 when no data remains.  
BIDI\_EINTR      Receive the signal. and the operation has been canceled.  
                    シグナル受信により処理が中断された  
BIDI\_ERROR      Other errors.  
                    その他のエラー

The Monitor MUST destroy the Bi-di Object when this function returns BIDI\_ERROR.

BIDI\_ERROR 発生時は、呼び元は bidiDestroy 関数によって Bi-di Object を破棄しなければならない。

The data stored in the buffer specified by *pData* is not guaranteed after this function returns BIDI\_EINTR. The Monitor MUST set again the appropriate data into the buffer before calling this function.

本関数を呼ぶ前に \*pData に格納したデータの内容は、BIDI\_EINTR 発生時は保証されない。正しい処理を行うためには再度 \*pData の内容を設定した後、本関数を呼ばなければならない。

# 4. Bi-di Plug-in Shared Library Interface

## 4.1 Corresponding functions

The following table defines the Bi-di Plug-in Shared Library Interface which has the corresponding functions to the Bi-di Plug-in API. The Bi-di Plug-in library MUST provide identical functions as the Bi-di Plug-in Shared Library Interface.

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

- The functions that the Bi-di Plug-in API must provide also MUST be provided by the Bi-di Plug-in library.
- The optional functions of the Bi-di Plug-in API that the Bi-di Plug-in library provides MUST be consistent with the return value of the `bidiLibGetCap()` function of the Bi-di Plug-in Shared Library Interface.

## 4.2 `bidiLibNew`

### Name

`bidiLibNew` – Create a new extended object that depends on each Bi-di Plug-in library.

### 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.

### Description

- This function provides the functionality of the `bidiNew()` function except loading and linking the Bi-di Plug-in library.
- This function MUST create an extended object that keeps a data set to control itself, and return a void pointer to it. Other Bi-di Plug-in Shared Library Interface functions receive the pointer to the extended object as a first argument.

### Return Value

Void pointer to the extended object or NULL when error.

## 4.3 Other functions

Each Bi-di Plug-in Shared Library Interface function except the `bidiLibNew()` function MUST be along the following specs.

### 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.

### Return Value

Returns the same value as the equivalent function of the Bi-di Plug-in API returns.



## 5. Bi-di Plug-in Process Interface

### 5.1 Pipe and Command Line Options

Bi-di Plug-in process and Stub exchange the printer command data and printer status data via the following four pipes.

process type の Bi-di Plug-in module と呼び元は、次の4つのパイプによりデータおよびコマンドを送受信する。

Data Writing Pipe	The pipe that the Stub sends the printer command data to the Bi-di Plug-in process. 呼び元から Bi-di Plug-in process へデータを送信するためのパイプ。
Data Reading Pipe	The pipe that the Bi-di Plug-in process sends the printer status data to the Stub. Bi-di Plug-in process から呼び元へプリンタステータスデータを送信するためのパイプ。
Command Writing Pipe	The pipe that the Stub send the request commands to the Bi-di Plug-in process. 呼び元から Bi-di Plug-in process へコマンドを送信するためのパイプ。
Command Reading Pipe	The pipe that the Bi-di Plug-in process send the acknowledge commands to the Stub. Bi-di Plug-in process から呼び元へコマンド応答を送信するためのパイプ。

- These pipes MUST be created by the Stub before the Stub. forks and executes the Bi-di Plug-in process.  
これらのパイプは、Bi-di Plug-in process の起動に先だって呼び元によって生成されなければならない。
- The Stub can obtain the file descriptor for the Data Reading Pipe by the `biDiGetReadFD()` function, and the file descriptor for the Data Writing Pipe by the `biDiGetWriteFD()` function.  
呼び元は Data Reading Pipe の file descriptor を `biDiGetReadFD` 関数で、Data Writing Pipe の file descriptor を `biDiGetWriteFD` 関数で取得する。
- The Command Writing Pipe and Command Reading Pipe are used inside the Stub and cannot be referred by the Monitor.  
Command Writing Pipe および Command Reading Pipe は呼び元からは直接は参照されず、Bi-di Plug-in API の内部処理において利用される。
- The Bi-di Plug-in process can refer each file descriptor by its arguments. The procedures in the `biDiNew()` function in the Stub MUST fork and execute the Bi-di Plug-in process with the following four arguments.  
Bi-di Plug-in process は、次のように process 起動時の引数によって、各パイプの file descriptor を受け取る。  
呼び元の内部処理では、次の4つの引数を全て指定して、Bi-di Plug-in process を起動しなければならない。

Command Line Option	Description	MUST or OPTIONAL
<code>--data-write-fd</code>	Data Writing Pipe file descriptor.	MUST
<code>--data-read-fd</code>	Data Reading Pipe file descriptor.	MUST
<code>--cmd-write-fd</code>	Command Writing Pipe file descriptor.	MUST
<code>--cmd-read-fd</code>	Command Reading Pipe file descriptor.	MUST
<code>--output-fd</code>	File descriptor to write the printer command data to the printer.	MUST
<code>--input-fd</code>	File descriptor to read the printer status data from the printer.	MUST

- Additionally, the procedures in the `biDiNew()` function in the Stub MAY pass the following argument to the Bi-di Plug-in process.  
さらに、呼び元は次の引数によって Bi-di Plug-in process に対してプリンタの URI を渡す場合がある。

Command Line Option	Description	MUST or OPTIONAL
<code>--printer-uri</code>	Device URI given to the <code>biDiNew()</code> function. If the <code>pURI</code> for the <code>biDiNew()</code> function is NULL, this argument MUST NOT be passed to the Bi-di Plug-in process.	OPTIONAL

- Spaces and one '=' character can be used for the delimiter between each command line option and file descriptor number.  
各引数と file descriptor number の間の区切り文字には、複数の空白文字および1つの '=' を許す。空白文字のみも可とする。  
example) `--data-write-fd = 5 --data-read-fd = 6 --cmd-write-fd = 7 --cmd-read-fd = 8 --output-fd = 4 input-fd = 4`
- The procedures in the `biDiNew()` function in the Stub MUST NOT pass the standard error file descriptor (2) for all the above options.  
呼び元は、上記6つ全ての file descriptor に 2(標準エラー出力)を指定してはいけなく、Bi-di Plug-in process は標準エラー出力に対してエラーメッセージを出力することができる。

## 5.2 Command Format

The command packet format for the Command Writing Pipe and Command Reading Pipe is below.

Command Writing Pipe および Command Reading Pipe 経由で送受信されるコマンドのフォーマットは次の通り。

Byte Offset	Description
0 - 3	Command ID
4 - 7	Data Length (=n bytes)
8 - 8 + (n-1)	Data

- Command ID and Data Length MUST be four bytes big-endian integer.  
Command ID および Data のバイト数は Big Endian の 4 バイト整数で指定する。
- Data MUST be big-endian and aligned on the one byte boundary.  
Data には Big Endian でデータを格納する。alignment は 1byte 境界とする。
- Command ID, Data and the request and acknowledge sequences for each command are defined for each Bi-di Plug-in API function. The Command IDs for each function are defined in the following table.  
Command ID、Data および送受信のシーケンスは、Bi-di Plug-in process の初期化処理および API の各々の関数毎に定義される。各々のコマンド ID と初期化処理、各 API の対応は次の通り。

Command ID	Value	Description
BIDI_CMD_NEW	0x00000001	Call the <code>bidisNew()</code> function.
BIDI_CMD_DESTROY	0x00000002	Call the <code>bidDestroy()</code> function.
BIDI_CMD_GETCAP	0x00000003	Call the <code>bidGetCap()</code> function.
BIDI_CMD_STARTJOB	0x00000011	Call the <code>bidStartJob()</code> function.
BIDI_CMD_ENDJOB	0x00000012	Call the <code>bidEndJob()</code> function.
BIDI_CMD_CANCELJOB	0x00000013	Call the <code>bidCancelJob()</code> function.
BIDI_CMD_STARTREAD	0x00000021	Call the <code>bidStartRead()</code> function.
BIDI_CMD_ENDREAD	0x00000022	Call the <code>bidRead()</code> function.
BIDI_CMD_READ	0x00000023	Call the <code>bidEndRead()</code> function.
BIDI_CMD_STARTWRITE	0x00000031	Call the <code>bidStartWrite()</code> function.
BIDI_CMD_ENDWRITE	0x00000032	Call the <code>bidWrite()</code> function.
BIDI_CMD_WRITE	0x00000033	Call the <code>bidEndWrite()</code> function.
BIDI_CMD_CTRL	0x00000034	Call the <code>bidCtrl()</code> function.

- The Command IDs for the acknowledge sequence are defined in the following table.  
また、上記コマンドへの応答に用いる Command ID として次のものを定義する。

Command ID	Value	Description
BIDI_CMD_OK	0x80000000	Success
BIDI_CMD_ERROR	0x80000001	Invalid command

- When the Bi-di Plug-in process detect an unknown or illegal command in the “Stub ==> Bi-di Plug-in process” request sequence for each command, the Bi-di Plug-in process MUST send the following command to the Stub in the next “Stub <== Bi-di Plug-in process” acknowledge sequence, and quit all remaining sequences for the command.  
後述の各々のコマンドの各々の Stub ==> Bi-di Plug-in process のシーケンスにおいて、Bi-di Plug-in process 側が未定義のコマンド等の不正コマンドを検出した場合は、それに続く Stub <== Bi-di Plug-in process のシーケンスにおいて、Bi-di Plug-in process 側は次のコマンドを Stub へ送出し、それ以降のシーケンスを中断しなければならない。  
Stub <== Bi-di Plug-in process  
Command ID = BIDI\_CMD\_ERROR  
Data Length = 0

## 5.3 BIDI\_CMD\_NEW

### Description

- Initialize and set up the properties of the Bi-di Plug-in process just after `biDiNew()` function forks and executes it.  
biDiNew 関数の実行に伴う Bi-di Plug-in process の起動直後の確認処理を行う。

### Sequence

- 1) The Bi-di Plug-in process receive the following command as soon as possible after being forked and executed.

Bi-di Plug-in process は起動後、出来るだけ早い時点で次のコマンドを Stub から受信する。

```
Stub ==> Bi-di Plug-in process
Command ID = BIDI_CMD_NEW
Data Length = 4
Data = Bi-di Plug-in version.
```

In the current version, the Stub MUST set 0x00010000 to the Bi-di Plug-in version.

Bi-di plug-in version には現時点では常に 0x00010000 を指定する。

- 2) After receiving the command in sequence “1”, the Bi-di Plug-in process compares the version received in sequence “1” and the version supported by the Bi-di Plug-in process. If the version in sequence “1” is less or equal to the version supported by the Bi-di Plug-in process, the Bi-di Plug-in process proceed to sequence “3”. If the version in sequence “1” is greater than the version supported by the Bi-di Plug-in process, the Bi-di Plug-in process MUST send the following command to the Stub, and idle until being killed by the Stub by the SIGTERM signal.

Bi-di Plug-in process は、1)のコマンドを受信後、その Data 中の version と Bi-di Plug-in process がサポートする version を比較する。Data 中の version Bi-di Plug-in process である場合は、3)以降の処理を継続する。

Data 中の version > Bi-di Plug-in process である場合は、次のコマンドを Stub に送信し、Stub から SIGTERM によって中断されるまで永久スリープしなければならない。

```
Stub <== Bi-di Plug-in process
Command ID = BIDI_CMD_OK
Data Length = 4
Data = BIDI_ERROR (Type:int, Length:4bytes)
```

- 3) The Bi-di Plug-in process initializes itself. If some error occurs, the Bi-di Plug-in process MUST send the following command to the Stub, and idle until being killed by the Stub by the SIGTERM signal.

Bi-di Plug-in process は、自身の初期化処理を継続し、その途中何らかのエラーが発生した場合は、次のコマンドを Stub に送信後、Stub から SIGTERM によって中断されるまで永久スリープしなければならない。

```
Stub <== Bi-di Plug-in process
Command ID = BIDI_CMD_OK
Data Length = 4
Data = BIDI_ERROR (Type:int, Length:4bytes)
```

- 4) After the initialization of the Bi-di Plug-in process, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は、その初期化処理が完了した後、次のコマンドを Stub に送信する。

```
Stub <== Bi-di Plug-in process
Command ID = BIDI_CMD_OK
Data Length = 4
Data = BIDI_OK (Type:int, Length:4bytes)
```

- 5) After receiving the command in sequence “4”, the Stub creates a new Bi-di Object, and returns the pointer to the Monitor.

Stub は 4)のコマンドを受信後、適切な Bi-di Object を生成して呼び元にそのポインタを返す。

- The Stub MUST wait at least 30 seconds for the command from the Bi-di Plug-in process. If the Stub cannot receive any commands from the Bi-di Plug-in process within 30 seconds, the Stub MUST kill the Bi-di Plug-in process by sending the SIGTERM to it, and clean up the Stub itself and return an error code to the Monitor.  
Stub は、Bi-di Plug-in process を起動後少なくとも 30 秒間、Bi-di Plug-in process からのコマンドの受信を待たなければならない。30 秒間待ってもコマンドが受信できない場合は、Bi-di Plug-in process を SIGTERM で kill し、後処理を行わなければならない。
- When the Stub receives the BIDI\_ERROR in “Data”, the Stub MUST send the SIGTERM signal to the Bi-di Plug-in program to kill it.  
Stub は、Bi-di Plug-in process から Data=BIDI\_ERROR を受信した場合は、Bi-di Plug-in process に対して SIGTERM を送信し、Bi-di Plug-in process の処理を中断しなければならない。

## 5.4 BIDI\_CMD\_DESTROY

### Description

- Execute the procedures for the `bidiDestroy()` function.  
`bidiDestroy` 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- When the Monitor calls the `bidiDestroy()` function, the Stub sends the following command to the Bi-di Plug-in process.  
`bidiDestroy` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process  
Command ID = BIDI\_CMD\_DESTROY  
Data Length = 0

- After receiving the command of sequence “1”, the Bi-di Plug-in process executes the termination procedures for itself, and after that, sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、終了処理を行ってから次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process  
Command ID = BIDI\_CMD\_OK  
Data Length = 0

And the Bi-di Plug-in process quits itself.

その後、Bi-di Plug-in process は process を終了する。

- After receiving the command in sequence “2”, the Stub clean up its properties, and return to the Monitor.  
Stub は 2) のコマンドを受信後、Bi-di Plug-in process 終了の後処理を行う。

## 5.5 BIDI\_CMD\_GETCAP

### Description

- Execute the procedures for the `bidiGetCap()` function.  
`bidiGetCap` 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- When the Monitor calls the `bidiGetCap()` function, the Stub sends the following command to the Bi-di Plug-in process.  
`bidiGetCap` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process  
Command ID = BIDI\_CMD\_GETCAP  
Data Length = 4  
Data = *cap* (Type:BiDiCap, Length:4bytes) : Argument *cap* of the `bidiGetCap()` function.

- After receiving the command of sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process  
Command ID = BIDI\_CMD\_OK  
Data Length = 4  
Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiGetCap()` function.

If the Bi-di Plug-in process receives an illegal command in sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.

1) で不正な *cap* 値を Bi-di Plug-in process が受信した場合は、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process  
Command ID = BIDI\_CMD\_OK  
Data Length = 4  
Data = BIDI\_ERROR (Type:int, Length:4bytes) : Return value of the `bidiGetCap()` function.

## 5.6 BIDI\_CMD\_STARTJOB

### Description

- Execute the procedures for the `bidiStartJob()` function.  
`bidiStartJob` 関数の実行を Bi-di Plug-in process に通知する。

## Sequence

- 1) When the Monitor calls the `bidiStartJob()` function, the Stub sends the following command to the Bi-di Plug-in process.  
bidiStartJob 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process

Command ID = BIDI\_CMD\_STARTJOB

Data Length = 4

Data = *idJob* (Type:int, Length:4bytes) : Argument *idJob* of the `bidiStartJob()` function.

- 2) After receiving the command of sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process

Command ID = BIDI\_CMD\_OK

Data Length = 4

Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiStartJob()` function.

## 5.7 BIDI\_CMD\_ENDJOB

### Description

- ・ Execute the procedures for the `bidiEndJob()` function.  
bidiEndJob 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- 1) When the Monitor calls the `bidiEndJob()` function, the Stub sends the following command to the Bi-di Plug-in process.  
bidiEndJob 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process

Command ID = BIDI\_CMD\_ENDJOB

Data Length = 0

- 2) After receiving the command of sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process

Command ID = BIDI\_CMD\_OK

Data Length = 4

Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiEndJob()` function.

## 5.8 BIDI\_CMD\_CANCELJOB

### Description

- ・ Execute the procedures for the `bidiCancelJob()` function.  
bidiCancelJob 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- 1) When the Monitor calls the `bidiCancelJob()` function, the Stub sends the following command to the Bi-di Plug-in process.

bidiCancelJob 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process

Command ID = BIDI\_CMD\_CANCELJOB

Data Length = 4

Data = *idJob* (Type:int, Length:4bytes) : Argument *idJob* of the `bidiCancelJob()` function.

- 2) After receiving the command of sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、ジョブの終了処理を行ってから次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process

Command ID = BIDI\_CMD\_OK

Data Length = 4

Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiCancelJob()` function.

## 5.9 BIDI\_CMD\_STARTREAD

### Description

- Execute the procedures for the `bidiStartRead()` function.  
`bidiStartRead` 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- 1) When the Monitor calls the `bidiStartRead()` function, the Stub sends the following command to the Bi-di Plug-in process.

`bidiStartRead` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process  
Command ID = BIDI\_CMD\_STARTREAD  
Data Length = 4 + 4 + n ()  
Data = `idReadMode` (Type:int, Length:4bytes) : Argument `idReadMode` of the `bidiStartRead()` function.  
n (Type:int, Length:4bytes) : Number of bytes of the argument `pLang` string.  
String (Type:char, Length:n bytes) : `pLang` string without a zero termination code.

- 2) After receiving the command of sequence “1”, the Bi-di Plug-in process prepares for sending the printer status data to the Stub, and sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、プリンタステータスデータの Stub への送出準備処理を完了してから、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process  
Command ID = BIDI\_CMD\_OK  
Data Length = 4  
Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiStartRead()` function.

The Bi-di Plug-in process establishes and retains the printer status data in the buffer and initializes the pointer for reading the data from the sequence “2” buffer. After that, the Bi-di Plug-in process MUST NOT update the printer status data until receiving the BIDI\_CMD\_ENDREAD command.

送出するプリンタステータスデータは 2) の時点で決定し、その read 開始位置を初期化する。以降 Bi-di Plug-in process は、BIDI\_CMD\_ENDREAD コマンドを受信するまで、プリンタステータスデータを更新してはならない。

## 5.10 BIDI\_CMD\_ENDREAD

### Description

- Execute the procedures for the `bidiEndRead()` function.  
`bidiEndRead` 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- 1) When the Monitor calls the `bidiEndRead()` function, the Stub sends the following command to the Bi-di Plug-in process.

`bidiEndRead` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

Stub ==> Bi-di Plug-in process  
Command ID = BIDI\_CMD\_ENDREAD  
Data Length = 0

- 2) After receiving the command of sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、自身のプリンタステータスデータ更新の再開処理を行ってから、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process  
Command ID = BIDI\_CMD\_OK  
Data Length = 4  
Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiEndRead()` function.

After sequence “2”, the Bi-di Plug-in process is permitted to update the printer status data in the buffer.

## 5.11 BIDI\_CMD\_READ

### Description

- Execute the procedures for the `bidiRead()` function.

biDiRead 関数の実行を Bi-di Plug-in process に通知し、プリンタステータスデータを Stub に送出する。

### **Sequence**

- 1) When the Monitor calls the biDiRead() function, the Stub sends the following command to the Bi-di Plug-in process.

biDiRead 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

```
Stub ==> Bi-di Plug-in process
Command ID = BIDI_CMD_READ
Data Length = 4
Data = nBufBytes (Type:int, Length:4bytes) : Argument nBufBytes of the biDiRead() function.
```

- 2) After receiving the command of sequence “1”, if the printer status data has already been established and retained in the buffer, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、プリンタステータスデータの送出準備が完了していた場合は、次のコマンドを Stub へ送信する。

```
Stub <== Bi-di Plug-in process
Command ID = BIDI_CMD_OK
Data Length = 4
Data = Number of bytes for sending the printer status data. (Type:int, Length:4bytes)
```

The bi-di Plug-in process compares the number of bytes of the printer status data that the Bi-di Plug-in process retains in the buffer and the number of bytes in “Data” received at sequence “1”, and sets the smaller one into the “Data” for sequence “2”. 送出する Byte 数は、1) で受信した nBufBytes と Bi-di Plug-in process が保持しているプリンタステータスデータのバイト数のうち、小さい方の値とする。

- 3) The bi-di Plug-in process reads the printer status data from the reading point of the buffer to the number of bytes in “Data” calculated at sequence “2”, and sends it to the Stub via the Data Reading Pipe.

Bi-di Plug-in process は、Data Reading Pipe 経由でプリンタステータスデータのうち read 開始位置から 2) で受信したバイト数分だけ Stub へ送信する。

- 4) After receiving the command of sequence “2”, the Stub receives the “Data” bytes printer status data from the Bi-di Plug-in process via the Data Reading Pipe.

Stub は 2) のコマンドを受信後 Data Reading Pipe 経由で 2) で受信したバイト数分のプリンタステータスデータを受信する。

## **5.12 BIDI\_CMD\_STARTWRITE**

### **Description**

- Execute the procedures for the biDiStartWrite() function.  
biDiStartWrite 関数の実行を Bi-di Plug-in process に通知する。

### **Sequence**

- 1) When the Monitor calls the biDiStartWrite() function, the Stub sends the following command to the Bi-di Plug-in process.

biDiStartWrite 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。

```
Stub ==> Bi-di Plug-in process
Command ID = BIDI_CMD_STARTWRITE
Data Length = 0
```

- 2) After receiving the command of sequence “1”, the Bi-di Plug-in process prepares for receiving the printer command data from the Stub, and sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、Stub からのデータ受信準備処理を完了してから、次のコマンドを Stub へ送信する。

```
Stub <== Bi-di Plug-in process
Command ID = BIDI_CMD_OK
Data Length = 4
Data = Return Value (Type:int, Length:4bytes) : Return value of the biDiStartWrite() function.
```

## **5.13 BIDI\_CMD\_ENDWRITE**

### **Description**

- Execute the procedures for the `bidiEndWrite()` function.  
`bidiEndWrite` 関数の実行を Bi-di Plug-in process に通知する。

### Sequence

- When the Monitor calls the `bidiEndWrite()` function, the Stub sends the following command to the Bi-di Plug-in process.  
`bidiEndWrite` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。  
 Stub ==> Bi-di Plug-in process  
 Command ID = BIDI\_CMD\_ENDREAD  
 Data Length = 0
- After receiving the command of sequence “1”, the Bi-di Plug-in process sends the following command to the Stub.  
 Bi-di Plug-in process は 1) のコマンドを受信後、次のコマンドを Stub へ送信する。  
 Stub <== Bi-di Plug-in process  
 Command ID = BIDI\_CMD\_OK  
 Data Length = 4  
 Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiEndWrite()` function.

## 5.14 BIDI\_CMD\_WRITE

### Description

- Execute the procedures for the `bidiWrite()` function.  
`bidiWrite` 関数の実行を Bi-di Plug-in process に通知し、Stub からデータを受信する。

### Sequence

- When the Monitor calls the `bidiWrite()` function, the Stub sends the following command to the Bi-di Plug-in process.  
`bidiWrite` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。  
 Stub ==> Bi-di Plug-in process  
 Command ID = BIDI\_CMD\_WRITE  
 Data Length = 4  
 Data = *nBufBytes* (Type:int, Length:4bytes) : Argument *nBufBytes* of the `bidiWrite()` function.
- After receiving the command of sequence “1”, if the Bi-di Plug-in process is ready to receive the printer command data, the Bi-di Plug-in process sends the following command to the Stub.  
 Bi-di Plug-in process は 1) のコマンドを受信後、データの受信準備が完了していた場合は、次のコマンドを Stub へ送信する。  
 Stub <== Bi-di Plug-in process  
 Command ID = BIDI\_CMD\_OK  
 Data Length = 0  
 Data = Number of bytes for receiving the printer command data. (Type:int, Length:4bytes)

The Bi-di Plug-in process compares the number of bytes of the printer command data that the Bi-di Plug-in process can receive and the number of bytes in “Data” received at sequence “1”, and sets the smaller one into the “Data” for sequence “2”.

- After receiving the command of sequence “2”, if the number of bytes in “Data” is greater than zero, the Stub sends the printer command data to the Bi-di Plug-in process to the number of bytes in “Data” via the Data Writing Pipe.
- If the number of bytes in “Data” of sequence “2” is greater than zero, the Bi-di Plug-in process receives the printer command data from the Stub to the number of bytes in “Data” via the Data Writing Pipe.

## 5.15 BIDI\_CMD\_CTRL

### Description

- Execute the procedures for the `bidiCtrl()` function.  
`bidiCtrl` 関数の実行を Bi-di Plug-in process に通知し、Stub からデータを受信する。

### Sequence

- When the Monitor calls the `bidiCtrl()` function, the Stub sends the following command to the Bi-di Plug-in process.  
`bidiCtrl` 関数の実行に伴い、Stub は次のコマンドを Bi-di Plug-in process へ送信する。  
 Stub ==> Bi-di Plug-in process  
 Command ID = BIDI\_CMD\_CTRL



Data Length = 8

Data = *idRequest* (Type:int, Length:4bytes) : Argument *idRequest* of the `bidiCtrl()` function.

*nDataBytes* (Type:int, Length:4bytes) : Argument *nDataBytes* of the `bidiCtrl()` function.

- 2) After receiving the command of sequence “1”, if the Bi-di Plug-in process is ready to receive the data, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 1) のコマンドを受信後、データの受信準備が完了していた場合は、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process

Command ID = BIDI\_CMD\_OK

Data Length = 0

- 3) The Stub sends the *nDataBytes* bytes data in the buffer specified by the argument *pData* of the `bidiCtrl()` function to the Bi-di Plug-in process via the Data Writing Pipe.

Stub は、Data Writing Pipe 経由でバッファ *pData* の *nDataBytes* のデータを Bi-di Plug-in process へ送出する。

- 4) After sending the command of sequence “2”, the Bi-di Plug-in process receives the *nDataBytes* bytes data from the Stub via the Data Writing Pipe.

Bi-di Plug-in process は 2) のコマンドを送信後 Data Writing Pipe 経由で *nDataBytes* のデータを受信する。

- 5) After receiving the data of sequence “4”) and executing the procedure specified by *idRequest*, the Bi-di Plug-in process sends the following command to the Stub.

Bi-di Plug-in process は 4) のデータを受信して *idRequest* に対応した処理をした後、次のコマンドを Stub へ送信する。

Stub <== Bi-di Plug-in process

Command ID = BIDI\_CMD\_OK

Data Length = 8

Data = Return Value (Type:int, Length:4bytes) : Return value of the `bidiCtrl()` function.

*nDataBytes* (Type:int, Length:4bytes) : Number of bytes of the return data stored in the *pData* buffer.

The value of *nDataBytes* in “Data” of the command of sequence “5”) might change from the value of *nDataBytes* in “Data” of the command of sequence “1”.

ここで得られる *nDataBytes* の値は 1) で与えられる *nDataBytes* の値と異なっている可能性がある。

- 6) The Bi-di Plug-in process sends the *nDataBytes* bytes data established at sequence “5”) via the Data Reading Pipe.

Bi-di Plug-in process は、Data Reading Pipe 経由で *nDataBytes* のデータを Stub へ送信する。尚、*nDataBytes* は 5) の値を参照する。

- 7) After receiving the command of sequence “5”, the Stub receives the *nDataBytes* bytes data established at sequence “5”) via the Data Reading Pipe, and store it into the buffer specified by *pData*.

Stub は 5) のコマンドを受信後 Data Reading Pipe 経由で *nDataBytes* のデータを受信してバッファ *pData* に格納する。尚、*nDataBytes* は 5) の値を参照する。

## 5.16 Signal Handling

The following table defines how the Bi-di Plug-in process SHOULD or MUST deal with each signal.

Process Type の Bi-di Plug-in process は、次のシグナルについて各々適切な処理を行う必要がある。

Signal	Processing
SIGTERM	<p>After receiving this signal, the Bi-di Plug-in process SHOULD terminate itself after terminating the printer status data reading procedure and printer command writing procedure. If the Bi-di Plug-in process receives this signal after receiving the BIDI_CMD_STARTJOB command and before dealing with the BIDI_CMD_ENJOB command, the termination procedure is equivalent to the procedure of dealing with the BIDI_CMD_CANCELJOB, BIDI_CMD_ENDJOB and BIDI_CMD_DESTROY commands in turn. If the Bi-di Plug-in process receives this signal before receiving the BIDI_CMD_STARTJOB command or after dealing with the BIDI_CMD_ENDJOB command, the termination procedure is equivalent to the procedure of dealing with the BIDI_CMD_DESTROY command. The Bi-di Plug-in process MUST NOT send any commands to the Stub after receiving this signal.</p> <p>プリンタステータスデータの read 処理、データの write 処理等の全ての処理を終了し、自分自身を終了する。その場合の終了処理は、BIDI_CMD_STARTJOB コマンドの受信後でかつ BIDI_CMD_ENDJOB コマンドの受信前であれば、BIDI_CMD_CANCELJOB、BIDI_CMD_ENDJOB、BIDI_CMD_DESTROY の3つのコマンドを1つずつ順番に受信した場合に実行される終了処理と等価の処理でなければならない。また、BIDI_CMD_STARTJOB コマンドの受信前であれば、BIDI_CMD_DESTROY コマンドを受信した場合に実行される終了処理と等価の処理でなければならない。尚、シグナル受信後は、Stub 側に一切のコマンドを送信してはならない。</p>
SIGPIPE	Same as above.
SIGHUP	<p>When receiving this signal during any request or acknowledge sequences of any commands, the Bi-di Plug-in process MUST cancel the command sequence and wait for a new command, such as BIDI_CMD_WRITE or BIDI_CMD_STARTJOB, etc, sent from the Stub.</p> <p>各々のコマンドの送受信シーケンスにおいて、シーケンスの途中の段階で本シグナルを受信した場合は、そのコマンドのシーケンスをリセットして、Stub 側からの BIDI_CMD_STARTJOB や BIDI_CMD_WRITE 等の新たなコマンドを待つ状態に復帰しなければならない。</p>

## 6. Printer Status Data Format

### 6.1 Printer Status Data XML Schema

The data format of the printer status data obtained from the `biDiRead()` function is defined in a set of XML schema files, including the Printer MIB v2 schema. These XML schema files are being considered in the PWG WBMM group, and currently located at the following FTP address.

<ftp://ftp.pwg.org/pub/pwg/wbmm/schemas/>

### 6.2 Printer Status Data Filtering

Obtaining the entire XML based printer status data periodically from the Bi-di Plug-in module may require a high data processing capability as well as a high data transfer rate. To reduce the amount of the printer status data, the Bi-di Plug-in API provides a filtering mechanism to obtain only the changed or variable printer status data.

When the Monitor calls the `biDiStartRead()` function, the printer status data is established and retained, and after that, the Monitor can obtain the printer status data from the `biDiRead()` function. The printer status data obtained from the `biDiRead()` function MUST be along the data specified by *idReadMode* given by the `biDiStartRead()` function prior to calling the `biDiRead()` function.

idReadMode	Description
BIDI_READ_PRT_MIB_ALL	Read all the printer status data.  The printer status data SHOULD include all the printer status data that the Bi-di Plug-in module supports. Initially, the Monitor obtains the entire printer status data from the Bi-di Plug-in module.
BIDI_READ_PRT_MIB_SUMMARY	Read the summarized printer status data.  The printer status data SHOULD include only the changed or variable data. The Monitor or upper system which receives the printer status data from the Monitor can detect the changed printer status data by comparing the printer status data obtained in the BIDI_READ_PRT_MIB_ALL mode with the data obtained in this mode.

## 7. Constants

### 7.1 Return Values

The following table defines the return value used in the Bi-di Plug-in API function definitions.

Define	Value
BIDI_OK	0
BIDI_TRUE	1
BIDI_FALSE	0
BIDI_ERROR	-1
BIDI_EINTR	-2
BIDI_EPROGRESS	-3
BIDI_ENOJOB	-4

## 8. Contributors and Author

---

### 8.1 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

### 8.2 Author

---

TORATANI Yasumasa	Canon Inc. 30-2, Shimomaruko 3-Chome, Ohta-ku, Tokyo 146-8501, Japan Email: <a href="mailto:toratani.yasumasa@canon.co.jp">toratani.yasumasa@canon.co.jp</a>
-------------------	--