

## ***SDK/J javax.usb Package***

### ***Developer's Guide***

*Version:1.0.6*



Copyright © 2011 Ricoh Co., Ltd.

### Terms of Use and Trademarks

1. The contents of this book may be changed without notice in the future.
2. The copying, reproducing, changing, quoting, reprinting, or distributing a part/all of this book are prohibited.
3. We make no warranty, express or implied, regarding this document and the sample codes described in this document. We will not be held responsible for any of our customer's losses, damages resulting from lost profits, or claims from any third party on using this document and the sample codes described in this document.
4. Trademarks

PostScript® and Acrobat® are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Red Hat is a registered trademark of Red Hat, Inc. in the United States and other countries.

Java, JVM (CVM) and CDC are trademarks or registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Eclipse is a trademark of eclipse.org in the United States, other countries, or both.

OSGi(TM) is a trademark, registered trademark, or service mark of The Open Services Gateway Initiative in the US and other countries.

Apache is a registered trademark of The Apache Software Foundation in the United States, other countries, or both.

Other product names used herein are for identification purposes only and might be trademarks of their respective companies. We disclaim any and all rights in those marks.

# Contents

<b>1. Introduction .....</b>	<b>2</b>
<b>2. Target readers .....</b>	<b>3</b>
<b>3. javax.usb SDK/J structure .....</b>	<b>4</b>
<b>4. Restrictions .....</b>	<b>5</b>
4.1. About the supported models.....	5
4.2. About the supported USB devices.....	5
4.2.1. The USB devices with which operation has been confirmed .....	5
4.2.2. The USB devices that are not supported .....	5
4.2.3. The number of devices that can be detected at one time .....	5
4.3. About the transfer modes .....	6
4.4. About the supported methods .....	6
4.5. About the coding.....	7
4.5.1. The coding to change the active UsbConfiguration/ UsbInterface .....	7
4.5.2. The coding for the interrupt transfer mode .....	7
4.5.3. Notes on the implementation on the MFP supporting USB slot on the front side .....	7
4.6. Changes for 2011 autumn and later models .....	8
4.6.1. Using with external keyboard .....	8
<b>5. Change history.....</b>	<b>9</b>

# 1. Introduction

This document describes javax.usb provided by Device SDK Type-J (hereinafter referred to as SDK/J).

javax.usb is an extended standard of the Java language and allows Java applications to access USB devices.

For the API specification of javax.usb, see the following document:

<http://javax-usb.cvs.sourceforge.net/javax-usb/javax-usb/docs/jsr80.pdf>

## 2. Target readers

This document is intended for application developers who wish to use javax.usb in SDK/J application development.

The reader is assumed to be familiar with:

- SDK/J application development
- javax.usb

### 3. javax.usb SDK/J structure

The picture below shows the javax.usb SDK/J structure.

SDK/J's javax.usb includes three Java packages: **jsr80.jar**, **jsr80\_ri.jar**, and **jsr80\_bsd.jar**.

These three packages are all necessary to form a complete functioning javax.usb API on the SDK/J platform.

Also, in addition to these three packages, the JNI for javax.usb is necessary.

#### **jsr80.jar**

The API of javax.usb.

#### **jsr80\_ri.jar**

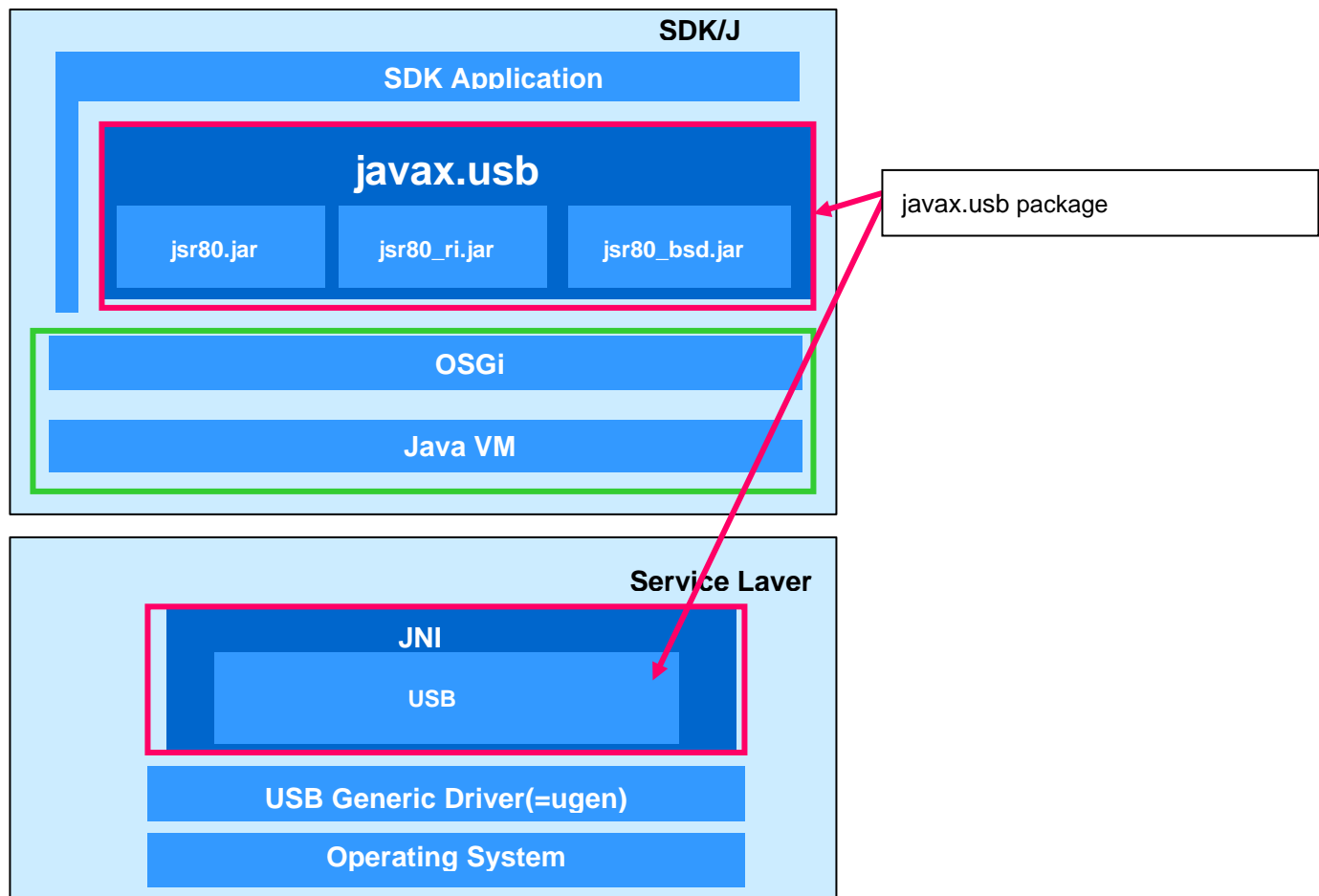
The common part of the OS-independent reference implementation.

#### **jsr80\_bsd.jar**

The reference implementation for the OS of MFPs/LPs.

#### **USB**

The JNI for javax.usb.



## 4. Restrictions

This section describes the restrictions on using javax.usb provided by SDK/J.

### 4.1. About the supported models

The following models are supported:

- The models are supported by SDK/J.
- The models support USB Host I/F by default.(Not Option Board)

### 4.2. About the supported USB devices

#### 4.2.1. The USB devices with which operation has been confirmed

See the sample application document " javaxusb\_sample\_eXXXX.doc".

#### 4.2.2. The USB devices that are not supported

The following USB devices are not supported:

- PC/SC-compliant USB devices

When the PC/SC daemon is running on the MFP/LP, a competitive situation will occur.

- PictBridge-compliant USB devices

A competitive situation will occur as the firmware of MFPs/LPs contains a module for PictBridge.

- USB hubs

\* The USB slot on the front side of the MFP is not supported since this slot is connected via USB hub with the USB slot on the rear side.

- USB devices using the isochronous transfer mode
- USB Mass Storage Class
- Composite device

#### 4.2.3. The number of devices that can be detected at one time

A maximum of four USB devices can be detected at one time.

If USB devices are connected to the MFP/LP beyond this limitation, they will not be detected.

### 4.3. About the transfer modes

Transfer Mode	IN	OUT	Note
Control	○	○	
bulk	○	○	
interrupt	○	✕	ugen Limitation
isochronous	✕	✕	ugen Limitation

### 4.4. About the supported methods

The following methods are not supported.

Class	Method	Note
UsbPipe	abortAllSubmissions ()	
	syncSubmit(byte[])	These methods will not be supported if: <b>a)</b> the value obtained by the getType() method is “javax.usb.UsbConst.ENDPOINT_TYPE_ISOCHRONOUS”, or <b>b)</b> the values obtained by the getType() method and the getDirection() method are “javax.usb.UsbConst.ENDPOINT_TYPE_INTERRUPT” and “javax.usb.UsbConst.ENDPOINT_DIRECTION_OUT” respectively.
	syncSubmit(List)	
	syncSubmit(UsbIrp)	
	asyncSubmit(byte[])	
	asyncSubmit(List)	
	asyncSubmit(UsbIrp)	



## 4.5. About the coding

### 4.5.1. The coding to change the active `UsbConfiguration/ UsbInterface`

Change of the active `UsbConfiguration/ UsbInterface` must be done before data transfer.

### 4.5.2. The coding for the interrupt transfer mode

In the interrupt transfer mode specification of this module, the buffer memory size is 1020 byte, and overflow data is lost. So, when coding an application, code it appropriately so that the buffer does not overflow. Specifically, when determining the intervals at which the `syncSubmit` method is called and the size of data to obtain, you should consider the values of “`wMaxPacketSize`” and “`bInterval`” of `UsbEndpointDescriptor`.

In case a buffer overflow has occurred, use the following method:

```
jp.co.ricoh.dsdk.util.usb.JavaxUsbUtil.reset
```

For details, see the `JavaDoc`.

### 4.5.3. Notes on the implementation on the MFP supporting USB slot on the front side

When activating the application using `javax.usb` on the MFP with USB slot on the front side, `javax.usb` detects the USB device attachment status (attach/detatch) even though `javax.usb` does not support the USB slot on the front side. Also, for the models which do not provide power to the USB slot of the front side during the energy-saving mode, the “detatch” event of the hub device may be sent to `javax.usb` even when there is no USB device connected to the front-side USB slot. The developers need to check which USB device has been attached or detached by using methods such as the `isUsbHub()` method.

## **4.6. Changes for 2011 autumn and later models**

### **4.6.1. Using with external keyboard**

When the external keyboard feature is being enabled, javax.usb cannot use USB devices for HID class. When using USB devices for HID class for javax.usb, disable the external keyboard feature. Changing the external keyboard setting requires operations by the service personnel.

## 5. Change history

Ver. 1.0.1beta1	Initial issue Implementation-Version 1.0.1-0.0
Ver. 1.0.1	Second issue Implementation-Version 1.0.1-1.0
Ver. 1.0.2	Third issue Implementation-Version 1.0.1-2.0 4.2.1. The USB devices with which operation has been confirmed
Ver. 1.0.3	4 <sup>th</sup> issue Implementation-Version 1.0.1-2.0 4.1. About the supported models
Ver. 1.0.4	5 <sup>th</sup> issue Implementation-Version 1.0.1-2.0 4.2.2. The USB devices that are not supported
Ver. 1.0.5	6th issue Implementation-Version 1.0.1-2.0 4.2.2. The USB devices that are not supported 4.5.3. Notes on the implementation on the MFP supporting USB slot on the front side
Ver. 1.0.6	7th issue Implementation-Version 1.0.1-2.0 4.2.2. The USB devices that are not supported 4.6. Changes for 2011 autumn and later models