TOSHIBA

TOSHIBA POS Terminal

ST-A10/ST-A20/ST-B10/ST-B20

Keylock Application User Manual
Toshiba TEC JavaPOS

Fourth Edition: June 16, 2010

TOSHIBA TEC CORPORATION

Revision Record

No. EAA-02564

Keylock Application User Manual

Rev. No.	Date	Pages	Description
---	Jul. 28, 2008	---	Original issue
1	Dec. 12, 2008	29-57	Descriptions about the PKB-ST5x have been added.
		1-29	In accordance with the addition of the descriptions described above, the chapter and section numbers have been renumbered.
		43,48,50	・Descriptions about checkHealth have been added. ・Descriptions about the key position have been changed. ・Setting information has been added.
		All	Review result has been reflected.
2	Jan, 20, 2010	1,29,49,50	Descriptions about the PKB-ST52 have been added.
3	Jun, 16, 2010	1	・Descriptions about Windows Embedded Point of Service1.1 (WEPOS1.1) have been added. ・Descriptions about SUSE Linux Enterprise Desktop 11 have been added.

Table of Contents

Introduction

The Keylock Application User Manual (hereinafter referred to as "this manual") was documented in accordance with the "UnifiedPOS Specifications Version 1.11" (hereinafter refered to as "UPOS Specification") which was published by the UnifiedPOS Committee for the purpose of standardization.
This manual mainly describes the specifications which are different from those described in the UPOS Specification and which are not described in it because they are Toshiba TEC's own specifications. For specifications not provided in this manual, please refer to the UPOS Specification.
Unless otherwise specified, this manual focuses on DeviceService.
The UPOS Specification can be downloaded from the following web site:
ARTS Home Page: http://www.nrf-arts.org/

Target Reader of This Manual

This document assumes that the reader is familiar with the following:

- General characteristics of POS peripheral devices
- General characteristics of Toshiba POS terminals and their peripheral devices
- General features of Windows and Linux
- Java terminology and architecture

Notes

Before reading this manual, please note the following:

- It is prohibited to use or duplicate a part or whole of this manual without the permission of Toshiba TEC Corporation.
- This manual is subject to change without prior notice.

Trademark Notification

* Windows, Windows 2000, Windows XP, and Windows Vista are registered trademarks of Microsoft Corporation in the United States and/or other countries.
The official name of Windows is the "Microsoft Windows Operating System".

* Linux is a registered trademark of Linus Torvalds.

* SUSE is a trademark of Novell.

* Java is a trademark of Sun Microsystems.

* iButton and 1-Wire are registered trademarks of Dallas Semiconductor Corp. Dallas Semiconductor is a wholly owned subsidiary of Maxim Integrated Products, Inc.

* All other product names mentioned in this document are trademarks or registered trademarks of their respective owners.

1. Keylock

1.1. Keylock JavaPOS Service ["TECKeylock"]

A class name of this Device Service is as follows:
“jpos.toshibatec.keylock”

1.1.1. Supported Operating Systems

This Device Service supports the following operating systems.

- Windows 2000
- Windows XP Professional
- Windows Embedded for Point of Service1.1(WEPOS1.1)
- Windows Embedded POSReady 2009
- Windows Vista
- SUSE Linux Enterprise Desktop10 SP1
- SUSE Linux Enterprise Desktop11

	Device	Windows 2000	Windows XP	WEPOS	Windows Vista	POSReady 2009	SUSE10	SUSE11
Default Keylock	KITST-A10-BTNK Electronic Keylock	A	A	A	A	N.A.	A	N.A.
PKBST-5x-Keylock	PKBST-50 Keylock(PS/2)	A	A	A	A	N.A.	A	N.A.
	LKBST-65 Keylock(PS/2)	A	A	A	A	N.A.	A	N.A.
	PKBST-52 Keylock(USB)	A.	A	A	A.	A	A.	A

1.1.2. Supported JavaVM

It is assumed this Device Service runs on the following JavaVM.

- Java2 SDK, Standard Edition Version 1.4.2

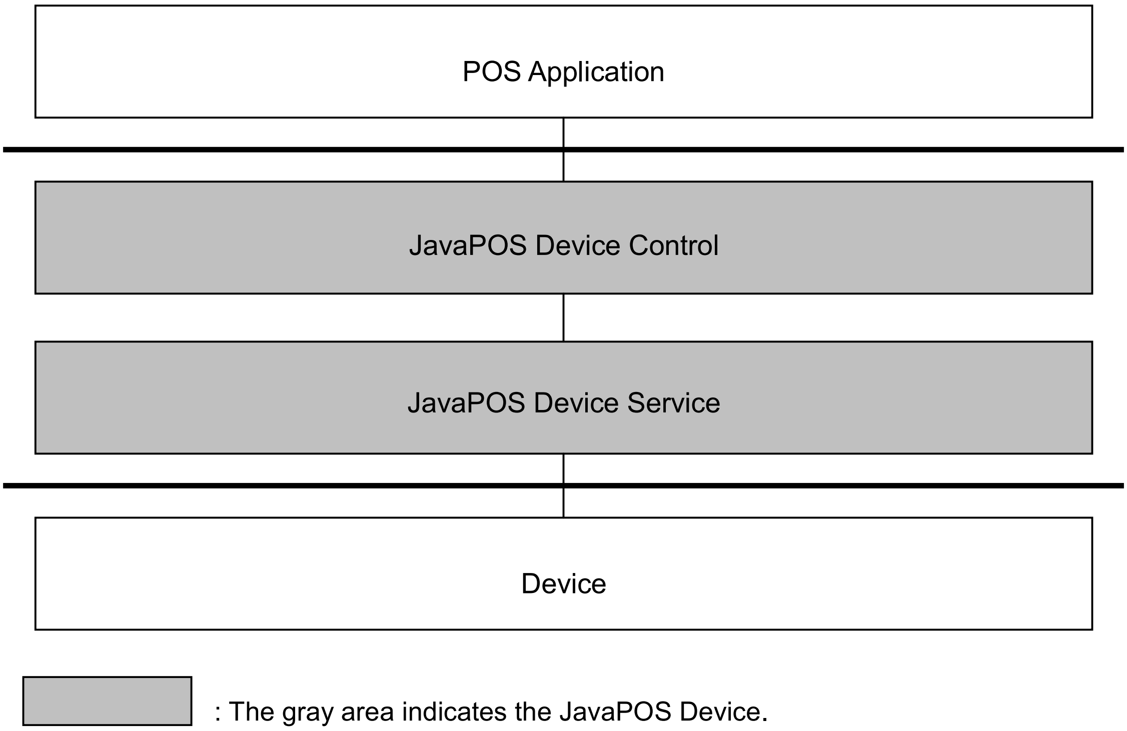
1.1.3. Supported Devices

This Device Service supports the following devices.

- KITST-A10-BTNK Electronic Keylock
  ST-A10 side: DS9490
  Key side: DS1990A
- Keylock device attached to the PKBST-50(PS/2)
- Keylock device attached to the PKBST-52(USB)
- Keylock device attached to the LKBST-65(PS/2)

1.1.4. Architecture Overview

The JavaPOS Device provides an Application Programming Interface (API) for the POS Application to communicate with the Device.



▨ : The gray area indicates the JavaPOS Device.

1.1.5. Property List

This Device Service provides the following properties which are in accordance with the UPOS Specification.

Common Property	Enable Condition	Description
DeviceControlDescription	None	Holds the Device Control name.
DeviceControlVersion	None	Holds the Device Control version number.
DeviceServiceDescription	open	Holds the Device Service name and the name of development company.
DeviceServiceVersion	open	Holds the Device Service version number.
PhysicalDeviceDescription	open	Holds the official Physical Device name.
PhysicalDeviceName	open	Holds the short Physical Device name.
CapCompareFirmwareVersion	open	Holds the status that indicates whether or not the function to compare the version of a firmware file against the firmware version of the Physical Device is provided.
CapPowerReporting	open	Holds the power reporting capability.
CapStatisticsReporting	open	Holds the status that indicates whether or not the statistics reporting function is provided.
CapUpdateFirmware	open	Holds the status that indicates whether or not the function to update the firmware file of the Device is provided.
CapUpdateStatistics	open	Holds the status that indicates whether or not the statistics update function is provided.
CheckHealthText	open	Holds the result of the most recent call to the checkHealth method.
Claimed	open	Holds the status whether or not the Device is claimed for exclusive access.
DeviceEnabled	open	Holds the status whether or not the Device is enabled.
FreezeEvents	open	Holds the status whether or not the event is frozen.
PowerNotify	open	Holds the power notification function type.
PowerState	open	Holds the current power condition.
State	None	Holds the current state of the JavaPOS Device.
Specific Property	**Enable Condition**	**Description**
CapKeyLockType	open	Holds the value that indicates the keylock type.
ElectronicKeyValue	open & enable	Holds the value read from the electronic keylock.
KeyPosition	open & enable	Holds the value that indicates the keylock position.
PositionCount	open	Holds the total number of keylock positions that are present on the keylock device.

Table 1 Keylock JavaPOS Device – Property List

1.1.6. Method List

The Device Service provides the following methods.

Common Property	Enable Condition	Description
open	None	
close	open	
claim	open	Not applicable
release	open & claim	Not applicable
checkHealth	open & enable	
compareFirmwareVersion	open & enable	Not supported
directIO	open	Not provided
resetStatistics	open & enable	Not supported
retrieveStatistics	open & enable	Not supported
updateFirmware	open & enable	Not supported
updateStatistics	open & enable	Not supported
Specific Property	**Enable Condition**	**Description**
waitForKeylockChange	open & enable	

Table 2 Keylock JavaPOS Device – Method List

1.1.7. Exception Specifications

This Device Service throws the following exception.
For the exception specifications, refer to the UPOS Specification.

Exception	Remarks
JposException	Method execution or a property setting failed.

Table 3 Keylock JavaPOS Device – Exception List

JposException Specifications

```
    Syntax:  JposException(int errorCode);
             JposException(int errorCode, int errorCodeExtended);
             JposException(int errorCode, String description);
             JposException(int errorCode, int errorCodeExtended, String description);
             JposException(int errorCode, String description, Exception origException);
             JposException(int errorCode, int errorCodeExtended, String description,
                                  Exception origException);
```

1.1.8. Log

This Device Service outputs a log using Commons Logging.
Depending on a situation, a log at either “INFO", "WARN", or "ERROR" level is output.

1.1.9. Log at INFO Level

A log at INFO level is output to check performance at a start and end of a method and a start and end of a property setting and when an event is thrown.

1.1.10. Log at WARN Level

A log at WARN level is output when an error, which is so minor that there is no need to throw an exception, has occurred. For example, the Device operates using a default value in the module because an unexpected value has been specified to jpos.xml.

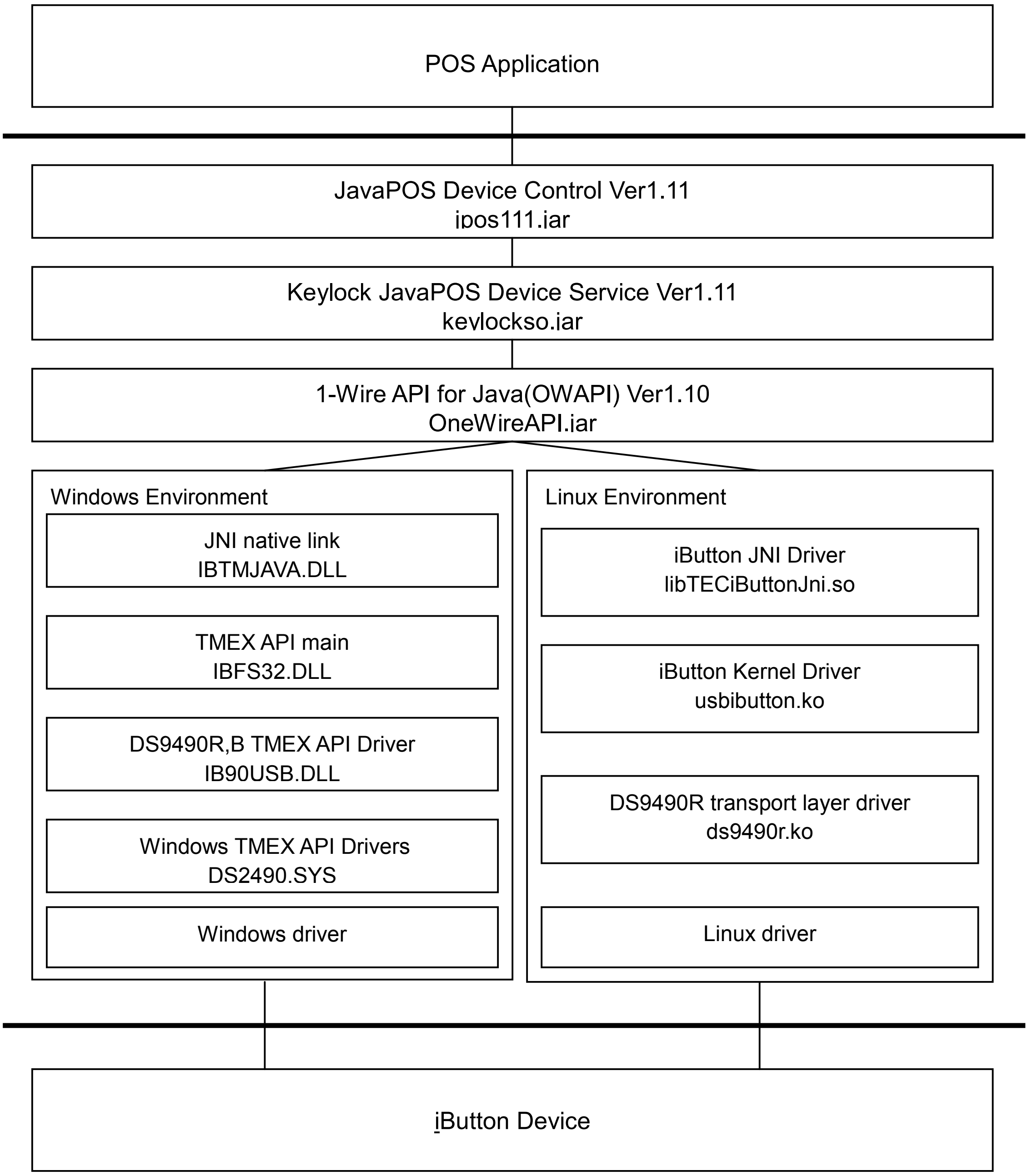
1.1.11. Log at ERROR Level

A log at ERROR level is output when a process in operation stops due to an error. Usually, this type of log is output when an exception is thrown.

1.2. iButton JavaPOS Device ["IBUTTON"]

1.2.1. Architecture Structure

The iButton JavaPOS Device uses some software to perform functions.
The software components shown below are required to build an execution environment.



1.2.2. Supported Functions

The functions supported/not supported by this Device Service are as follows:

Supported Functions	Unsupported Functions
Electronic key	Power status notification Accumulation of statistics Update of statistics Update of firmware Comparison of firmware version
Extended Functions	
None	

Table 4 Keylock JavaPOS Device – Functions

1.2.3. Property Specifications

Property List

This Device Service provides the following properties.

For details of property specifications, refer to the section, “1.2.3.1 Details of Properties” and the UPOS Specification.

Common Property	**Enable Condition**	**Description**
DeviceControlDescription	None	Holds the Device Control name. **(*1)**
DeviceControlVersion	None	Holds the Device Control version number. **(*1)**
DeviceServiceDescription	open	Holds the Device Service name and the name of development company. **(*1)**
DeviceServiceVersion	open	Holds the Device Service version number. **(*1)**
PhysicalDeviceDescription	open	Holds the official Physical Device name. **(*1)**
PhysicalDeviceName	open	Holds the short Physical Device name. **(*1)**
CapCompareFirmwareVersion	open	Holds the status that indicates whether or not the function to compare the version of a firmware file against the firmware version of the Physical Device is provided. **(*1)**
CapPowerReporting	open	Holds the power reporting capability. **(*1)**
CapStatisticsReporting	open	Holds the status that indicates whether or not the statistics reporting function is provided. **(*1)**
CapUpdateFirmware	open	Holds the status that indicates whether or not the function to update the firmware file of the Device is provided. **(*1)**
CapUpdateStatistics	open	Holds the status that indicates whether or not the statistics update function is provided. **(*1)**
CheckHealthText	open	Holds the result of the most recent call to the checkHealth method. For details, refer to the section, “1.2.5.2 Details of Methods”.
Claimed	open	Holds the status whether or not the Device is claimed for exclusive access. **(*1)**
DeviceEnabled	open	Holds the status whether or not the Device is enabled.
FreezeEvents	open	Holds the status whether or not the event is frozen.
PowerNotify	open	Holds the power notification function type. **(*1)**
PowerState	open	Holds the current power condition. **(*1)**
State	None	Holds the current state of the JavaPOS Device.
Specific Property	**Enable Condition**	**Description**
CapKeyLockType	open	Holds the value that indicates the keylock type. **(*1)**
ElectronicKeyValue	open & enable	Holds the value read from the electronic keylock.
KeyPosition	open & enable	Holds the value that indicates the keylock position. **(*1)**
PositionCount	open	Holds the total number of keylock positions that are present on the keylock device. **(*1)**

(*1) For these properties, a specified value is set when the open method is executed and will not be updated until a next open method. For details of the values to be set, refer to the table, “Table 6 Keylock JavaPOS Device – Property Value List (in part)”.

Table 5 Keylock JavaPOS Device – Property List

This table only lists the device-dependent properties

Common Property	**Value**
DeviceControlDescription	“JavaPOS Keylock Device Control”
DeviceControlVersion	“1011000”
DeviceServiceDescription	“TEC JavaPOS Keylock Device Service”
DeviceServiceVersion	“1011XXX” **(*1)**
PhysicalDeviceDescription	“TEC {DS9404} Electronic Keylock"
PhysicalDeviceName	“TEC {DS9404}”
CapCompareFirmwareVersion	false
CapPowerReporting	JPOS_PR_NONE
CapStatisticsReporting	false
CapUpdateFirmware	false
CapUpdateStatistics	false
CapKeylockType	LOCK_KT_ELECTRONIC
Claimed	false **(*2)**
PowerNotify	JPOS_PN_DISABLED **(*2)**
PowerState	JPOS_PS_UNKNOWN **(*2)**
Specific Property	**Value**
KeyPosition	0 **(*3)**
PositionCount	0

(*1) Build version is indicated as “XXX” because this manual may not be revised as soon as the module is updated.

(*2) Due to the limitations of the Device, the value of these properties does not change.

(*3) The KeyPosition property value is always “0” if the keylock device is of electronic type.

Table 6 Keylock JavaPOS Device – Property Value List (in part)

1.2.3.1. Details of Properties

DeviceControlDescription Property

Type

String DeviceControlDescription;

Mutability

Read Only

Remarks

Holds the Device Control name.

The Device returns the following character string:

“TEC JavaPOS Keylock Device Service”

This property is always readable.

Exception

When this property is accessed, no exceptions are thrown.

DeviceControlVersion Property

Type

String DeviceControlVersion();

Mutability

Read Only

Remarks

Holds the Device Control version number.

The Device returns the following version number:

1011000

This number indicates the Device Control version number is 1.11.0 (in accordance with the JPOS Specification Version 1.11). Here, Major no. is 1, Minor number is 11, and Build no. is 0.

This property is always readable.

Exception

When this property is accessed, no exceptions are thrown.

DeviceServiceDescription Property

Type

String DeviceServiceDescription;

Mutability

Read Only

Remarks

Holds the Device Service name.
The Device returns the following character string.
"JavaPOS Keylock Device Control"
This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

DeviceServiceVersion Property

Type

String DeviceServiceVersion;

Mutability

Read Only

Remarks

Holds the Device Service version number.
The Device returns the following version number.
1011XXX
This number indicates the Device Service version number is 1.11.XXX (in accordance with the JPOS Specification Version 1.11) (The value, "XXX" indicates a build version, which is incremented from 000.). Here, Major no. is 1, Minor number is 11, and Build no. is XXX.
This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

PhysicalDeviceDescription Property

Type

String PhysicalDeviceDescription;

Mutability

Read Only

Remarks

Holds the official Physical Device name.
The Device returns the following character string.
"TEC {DS9404} Electronic Keylock"
This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

PhysicalDeviceName Property

Type

String PhysicalDeviceName;

Mutability

Read Only

Remarks

Holds the short Physical Device name.
The Device returns the following character string.

"TEC {DS9404}"

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapCompareFirmwareVersion Property

Type

boolean CapCompareFirmwareVersion;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the function to compare the version of a firmware file against the firmware version of the Physical Device is provided.

Value	Meaning
true	This function is supported.
false	This function is not supported.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapPowerReporting Property

Type

int CapPowerReporting;

Mutability

Read Only

Remarks

Holds the power reporting capability.

Value	Meaning
JPOS_PR_NONE	Power reporting function is not available because the state of the Device cannot be determined.
JPOS_PR_STANDARD	Two types of power state (OFF_OFFLINE, ONLINE) can be determined and reported.
JPOS_PR_ADVANCE	Three types of power state (OFF, OFFLINE, ONLINE) can be determined and reported.

The Device returns JPOS_PR_NONE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapStatisticsReporting Property

Type

boolean CapStatisticsReporting;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the statistics reporting function is provided.

Value	Meaning
true	This function is provided.
false	This function is not provided.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapUpdateFirmware Property

Type

boolean CapUpdateFirmware;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the function to update the firmware file of the Device is provided.

Value	Meaning
true	This function is provided.
false	This function is not provided.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapUpdateStatistics Property

Type

boolean CapUpdateStatistics;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the statistics update function by the Device Service or the Device is provided.

Value	Meaning
true	This function is provided.
false	This function is not provided.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CheckHealthText Property

Type

String CheckHealthText;

Mutability

Read Only

Remarks

Holds the result of the most recent call to the **checkHealth** method. For details, refer to the description of the **checkHealth** method in the section, “1.2.4.2 Details of Method”.

When the **checkHealth** method has not been called yet, “” (empty string) is held for this property.

Exception

When this property is accessed, no exceptions are thrown.

Claimed Property

Type

boolean Claimed;

Mutability

Read Only

Remarks

Holds the status whether or not the Device is claimed for exclusive access.

Value	Meaning
true	The Device is claimed for exclusive access.
false	The Device is released.

This property always returns FALSE because the Device does not provide the exclusive access function.

Exception

When this property is accessed, no exceptions are thrown.

DeviceEnabled Property

Type

boolean DeviceEnabled;

Mutability

Read / Write

Remarks

Holds the status whether or not the Device is enabled.

Value	Meaning
true	The Device is enabled (in an operational state).
false	The Device is disabled.

By setting this property to TRUE, the Device is enabled (in an operational state) and to FALSE, the Device is disabled.

This property is initialized to FALSE by the **open** method.

Exception

While setting this property, a **JposException** is thrown. For details, refer to the section, “1.2.6.1Exceptions Thrown by Property Setting”.

FreezeEvents Property

Type

boolean FreezeEvents;

Mutability

Read / Write

Remarks

Holds the status whether or not the event is frozen.

Value	Meaning
true	Events are frozen, which means they are queued and not delivered.
false	Events are delivered.

If TRUE, events are frozen, that is, they are queued and not delivered. By setting this property to FALSE, events are allowed to be delivered (if other event delivery conditions are met). If there are some events which have been held in a **FreezeEvents** state, changing this property to FALSE will allow these events to be delivered (if other event delivery conditions are met).

If an interruption by an event is not desirable, the application can choose whether or not the event is to be frozen.

This property is initialized to FALSE by the **open** method.

Exception

While setting this property, a **JposException** is thrown. For details, refer to the section, “1.2.6.1 Exceptions Thrown by Property Setting”.

PowerNotify Property

Type

int PowerNotify;

Mutability

Read / Write

Remarks

Holds the power notification function type.

Value	Meaning
JPOS_PN_DISABLED	Power state is not reported. Neither the StatusUpdateEvent event for power notification is delivered nor power state is held in the PowerState property.
JPOS_PN_ENABLED	When the DeviceEnabled property is TRUE, the StatusUpdateEvent event for power notification is delivered and the power state is held in the PowerState property.

Whenever the **open** method is executed, JPOS_PN_DISABLED is always returned because the Device does not provide the power notification function.

Exception

While setting this property, a **JposException** is thrown. For details, refer to the section, "1.2.6.1 Exceptions Thrown by Property Setting".

PowerState Property

Type

int PowerState;

Mutability

Read Only

Remarks

Holds the current power condition.

Value	Meaning
JPOS_PS_UNKNOWN	Power state cannot be determined.
JPOS_PS_ONLINE	The Device power is on and the Device is in ready state.
JPOS_PS_OFF	The Device power is off or the Device is not connected.
JPOS_PS_OFFLINE	The Device power is on, but "not ready state" cannot be returned in response to a request.
JPOS_PS_OFF_OFFLINE	The Device is off or in offline state.

Whenever the **open** method is executed, JPOS_PS_UNKNOWN is always returned because the Device does not provide the power notification function.

Exception

When this property is accessed, no exceptions are thrown.

State Property

Type

int State;

Mutability

Read Only

Remarks

Holds the current state of the JavaPOS Device.

Value	Meaning
JPOS_S_CLOSED	The Device is closed.
JPOS_S_IDLE	The Device is in a normal state and is not busy.
JPOS_S_BUSY	The Device is in a normal state and is busy performing output.
JPOS_S_ERROR	An error has been reported. To resume I/O, the Device must be recovered to a normal state.

Whenever the **open** method is executed, JPOS_ S_IDLE is always returned.

Exception

When this property is accessed, no exceptions are thrown.

CapKeylockType Property

Type

int CapKeylockType;

Mutability

Read Only

Remarks

Holds the value that indicates the keylock type.

Value	Meaning
LOCK_KT_STANDARD	Standard keylock. (Compatible with the JPOS Version 1.10 or earlier)
LOCK_KT_ELECTRONIC	Electronic keylock. Holds the value read from the electronic keylock using the **ElectronicKeyValue** property. Always set to “0” for the **PositionCount** and **KeyPosition** properties.

The Device returns LOCK_KT_ELECTRONIC.

This property is initialized to FALSE by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

ElectronicKeyValue Property

Type

byte[] ElectronicKeyValue;

Mutability

Read Only

Remarks

Holds the value read from the electronic key.

If **DeviceEnabled** is TRUE, this property is initialized to null, and the value is stored when the electronic key is touched or removed. For details of the values, refer to the section, 1.2.8 Limitations and Precautions, 3) Electronic key data”.

Exception

When this property is accessed, no exceptions are thrown.

KeyPosition Property

Type

int KeyPosition;

Mutability

Read Only

Remarks

Holds the value that indicates the keylock position.

The **StatusUpdateEvent** event is delivered when the keylock position is changed. Accordingly, this property value is updated. This property has one of the following values:

Value	Meaning
LOCK_KP_LOCK	Keylock is in the “locked” position.
LOCK_KP_NORM	Keylock is in the “normal” position.
LOCK_KP_SUPR	Keylock is in the “supervisor” position.
Other values	Keylock is in the position other than the above.

Whenever the **open** method is executed, “0” is always returned because the Device is of electronic keylock type.

Exception

When this property is accessed, no exceptions are thrown.

PositionCount Property

Type

int PositionCount;

Mutability

Read Only

Remarks

Holds the total number of keylock positions that are present on the keylock device.

Whenever the **open** method is executed, “0” is always returned because the Device is of electronic keylock type.

Exception

When this property is accessed, no exceptions are thrown.

1.2.4. Method Specifications

1.2.4.1. Method List

The Device Service provides the following methods.

For the specifications of the checkHealth and directIO methods, refer to the descriptions of these methods in the section, “1.2.4.2 Details of Methods, and for those of other methods, refer to the UPOS Specification.

Common Method	Requirement	Remarks
open	None	
close	open	
claim	open	Not applicable
release	open & claim	Not applicable
checkHealth	open & enable	
compareFirmwareVersion	open & enable	Not supported
directIO	open	Not provided
resetStatistics	open & enable	Not supported
retrieveStatistics	open & enable	Not supported
updateFirmware	open & enable	Not supported
updateStatistics	open & enable	Not supported
Specific Method	**Requirement**	**Remarks**
waitForKeylockChange	open & enable	

Table 7 Keylock JavaPOS Device – Method List

1.2.4.2. Details of Methods

open Method

Type

void open(String logicalDeviceName **) throws JposException;**

Parameter	Description
logicalDeviceName	Specifies a name of the Device to be opened. An attribute value specified by the attribute name “logicalName” in the <JposEntry> element under the <JposEntries> element of the jpos.xml file must be specified to this value. For this Device, “IBUTTON” must be specified.

Remarks

This method is called to open the Device for subsequent I/O.

Exception

When this method is invoked, a **JposException** is thrown. For details, refer to the section “1.2.6.1 Exceptions Thrown by Methods”.

close Method

Type

void close() throws JposException;

Remarks

Releases the Device and its resources.

If the **DeviceEnabled** property is TRUE, the Device is disabled.

Exception

When this method is invoked, a **JposException** is thrown. For details, refer to the section “1.2.6.1 Exceptions Thrown by Methods”.

claim Method

Type

void claim(int timeout **) throws JposException;**

Parameter	Description
timeout	Specifies the maximum wait time in milliseconds to obtain an exclusive access. If “0” is set and the exclusive access is obtained, the method immediately returns the control to the application. For other cases, a JposException is thrown. If JPOS_FOREVER (-1), this method waits as long as needed until the exclusive access is obtained.

Remarks

This method is called when an exclusive access to the Device is requested.

For this Device, a JposException is always thrown because it cannot obatin the exclusive access.

Exception

When this method is invoked, a **JposException** is thrown. For details, refer to the section “1.2.6.1 Exceptions Thrown by Methods”.

release Method

Type

void release () throws JPOSException;

Remarks

This method is called to release an exclusive access to the Device.

For this Device, a JposException is always thrown because it cannot obatin the exclusive access.

Exception

When this method is invoked, a **JposException** is thrown. For details, refer to the section “1.2.6.1 Exceptions Thrown by Methods”.

checkHealth Method

Type

void checkHealth(int level **) throws JposException;**

Parameter	Description
level	Specifies the type of health check to be performed on the Device.

Any of the following values can be specified for the level parameter.

The *Level* parameter indicates the following

Value	Meaning
JPOS_CH_INTERNAL	Performs a health check that does not physically use the device.
JPOS_CH_EXTERNAL	Performs a thorough test using the Device.
JPOS_CH_INTERACTIVE	Performs an interactive test with the Device.

Remarks

This Device Service supports a health check of the External and Interactive levels. Regardless of level, the following exceptions are thrown:

Exception’s ErrorCode	CheckHealthText Property	Meaning
JPOS_E_CLOSED	No change	The Device has been closed.
JPOS_E_DISABLED	“HCheck:Disabled”	The Device has been disabled.
JPOS_E_ILLEGAL	“HCheck:Illegal”	Illegal level parameter
JPOS_E_FAILURE	“HCheck:failure”	Captures an exception other than JavaPOS.

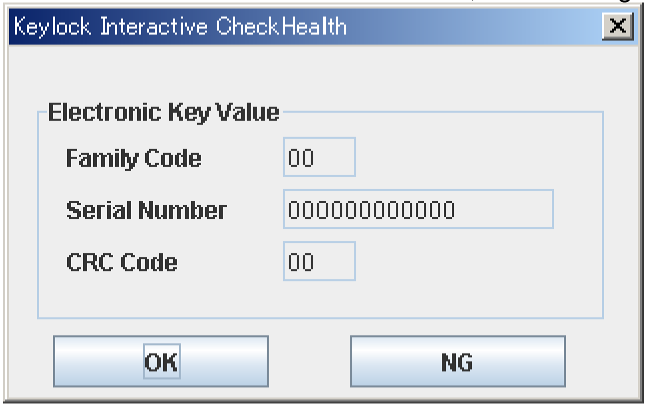
1) Internal Level (level=JPOS_CH_INTERNAL)
This Device Service does not support the checkHealth method at an Internal level.
The checkHealth method at an Internal level throws the following exception.

Exception's ErrorCode	CheckHealthText Property	Meaning
JPOS_E_ILLEGAL	"Internal HCheck:Illegal"	Not supported

2) External Level (level=JPOS_CH_EXTERNAL)
Checks the electronic key is connected.
The checkHealth method at an External level throws the following exceptions.

Exception's ErrorCode	CheckHealthText Property	Meaning
– (No exceptions are thrown.)	"External HCheck:Successful"	The electronic key exists.
JPOS_E_FAILURE	"External HCheck:No Hardware"	The electronic key does not exist.
JPOS_E_ILLEGAL	"External HCheck:Driver Error"	Driver error

3) Interactive Level (level=JPOS_CH_INTERACTIVE)
When the checkHealth method at an Interactive level is invoked, the following dialog box is displayed.


Keylock Interactive CheckHealth

Electronic Key Value

Family Code 00

Serial Number 000000000000

CRC Code 00

When the iButton is inserted after the dialog box appears, the electronic key data stored in the iButton device is read and the value for Family Code, Serial Number, and CRC Code are displayed. These values disappear when the electronic key is removed and these boxes are kept blank until the electronic key is inserted again.
After visually checking the values, press the "OK" button if the values are correct, or "NG" if there is any error.
By pressing the "OK" or "NG" button, the dialog box is closed and the control is returned to the application.

The checkHealth method at an Interactive level throws the following exceptions.

Exception's ErrorCode	CheckHealthText Property	Meaning
– (No exceptions are thrown.)	"Interactive HCheck:Successful"	Completed with the "OK " button
JPOS_E_FAILURE	"Interactive HCheck:Error"	Completed with the "NG " button

Exception

When this method is invoked, a **JposException** is thrown. For details, refer to the section "1.2.6.1 Exceptions Thrown by Methods".

directIO Method

Type

void directIO(int command, **int[]** data, **Object** object **) throws JposException;**

Parameter	Meaning
command	Specifies a command number.
data	An array of one mutable integer of which specific values vary.
object	Specifies an additional data.

The command, data and object parameter values vary depending on the Device Service.

Remarks

This method provides the Device functions, which are not supported by the method and properties, to the application.

For this Device, a JposException is always thrown because there are no functions which this method can provide.

Exception

When this method is invoked, a **JposException** is thrown. For details, refer to the section “1.2.6.1 Exceptions Thrown by Methods”.

1.2.4.3. directIO Method Specifications

Syntax: directIO(int command, int[] data, Object object) throws JposException;

This Device Service provides no functions that can be executed by the directIO method.

Regardless of command parameter value, the following exceptions are thrown by the directIO method.

Exception's ErrorCode	CheckHealthText Property	Meaning
JPOS_E_CLOSED	0	Not open
JPOS_E_ILLEGAL	0	Not supported

1.2.5. Event Specifications

This Device Service throws the following events.
For the event specifications, refer to the UPOS Specification.

Event	Remarks
StatusUpdateEvent	Notifies when the Device’s operation status has changed.

Table 8 Keylock JavaPOS Device – Event List

1.2.5.1. StatusUpdateEvent Event Specifications

Syntax: StatusUpdateEvent(int Status);

This Device Service throws the following StatusUpdateEvent event.

Status Parameter	Meaning
LOCK_KP_ELECTRONIC	iButton is either inserted or removed.

1.2.6. Exception Specifications

This Device Service throws the following exception.
For the exception specifications, refer to the UPOS Specification.

Exception	Remarks
JposException	Method execution or a property setting failed.

Table 9 Keylock JavaPOS Device – Exception List

JposException Specifications

```
Syntax:  JposException(int errorCode);
         JposException(int errorCode, int errorCodeExtended);
         JposException(int errorCode, String description);
         JposException(int errorCode, int errorCodeExtended, String description);
         JposException(int errorCode, String description, Exception origException);
         JposException(int errorCode, int errorCodeExtended, String description,
                              Exception origException);
```

1.2.6.1. Exceptions Thrown by Methods

This Device Service throws the following exceptions when methods are invoked.

Method	Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
open	JPOS_E_ILLEGAL,0	The Device is open.	The Device has been open.
	JPOS_E_NOEXIST,0	The Device was not found.	DeviceService was not found. Check the logicalDeviceName parameter value and the jpos.xml setting.
	JPOS_E_NOSERVICE,0	DeviceService is illegal.	DeviceService does not support the methods which it should. Or, the version number is invalid. Check the correct module is used.
close	JPOS_E_CLOSED,0	The Device has been closed.	—
claim	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This Device Service does not support this function.
release	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.
checkHealth	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_DISABLED,0	The Device is disabled.	Try again after setting the DeviceEnable property to TRUE.
	JPOS_E_ILLEGAL,0	An invalid parameter value was specified.	Check the specified value and specify a valid one.
		Unsupported function was specified.	This function is not supported.
	JPOS_E_FAILURE,0	A health check failed.	The health check failed as a result of visual check or an internal error. Try again after checking the Device. If the error persists, investigate the error.
compareFirmwareVersion	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.
directIO	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.
resetStatistics	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.
retrieveStatistics	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.
updateFirmware	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.
updateStatistics	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

Method	Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
waitForKeylockChange	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_DISABLED,0	The Device is disabled.	Try again after setting the DeviceEnable property to TRUE.
	JPOS_E_ILLEGAL,0	An invalid parameter value was specified.	Check the specified value and specify a valid one.
	JPOS_E_TIMEOUT,0	A specified timeout period expired.	A status of the iButton did not change within a period of time specified by the timeout parameter.

1.2.6.2. Exceptions Thrown by Property Setting

This Device Service throws the following exceptions when property settings are performed.

Method	Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
setDeviceEnabled	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_FAILURE,0	Enabling the Device failed.	Enabling the Device failed. Try again after checking a status of the Device. If the error persists, investigate the error.
setFreezeEvents	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
setPowerNotify	JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
	JPOS_E_ILLEGAL,0	An invalid parameter value was specified.	Check the specified value and specify a valid one.
		The Device is enabled.	Try again after setting the DeviceEnable property to FALSE.
		Unsupported function was specified.	This function is not supported.

1.2.7. Setting Information

Setting information of this Device Service is set in the XML file called "jpos.xml".
The <prop> tag in the XML file is a setting item specific to this Device. For details of other tags, <creation>, <vendor>, <jpos>, and <product>, refer to the UPOS Specification.

```
<JposEntries>
        < JposEntry logicalName="IBUTTON">
        <JposEntry logicalName="IBUTTON">
            <creation factoryClass="jpos.toshibatec.loader.JavaPOSServiceFactory"
                serviceClass="jpos.toshibatec.services.KeylockService"/>
            <vendor name="TOSHIBA TEC Corporation" url="http://www.toshibatec.co.jp"/>
            <jpos category="Keylock" version="1.11"/>
            <product description=" TEC i-Button Electronic Keylock "
                name=" TEC i-Button " url="http://www.toshibatec.co.jp"/>

            <prop name="modelName" type="String" value="{DS9490}"/>
            <prop name="FamilyCodes" type="String" value="01"/>
            <prop name="deviceBus" type="String" value="USB"/>
        </JposEntry>
```

Item Name	Value
modelName	Specify the model number of the iButton device. Specify "{DS9490}" for ST-A10 and ST-A20.
FamilyCodes	Specify the family codes of the iButton key to be used. More than one codes can be specified using commas “,”. Specify “01” for ST-A10 and ST-A20.
deviceBus	Specify the method of connection with the Device. Specify "USB" for ST-A10 and ST-A20.

Table 10 Keylock JavaPOS Device – Setting Information List

1.2.8. Limitations and Precautions

This section describes the limitations and precautions for using this Device Service, including the differences from the UPOS Specifications.

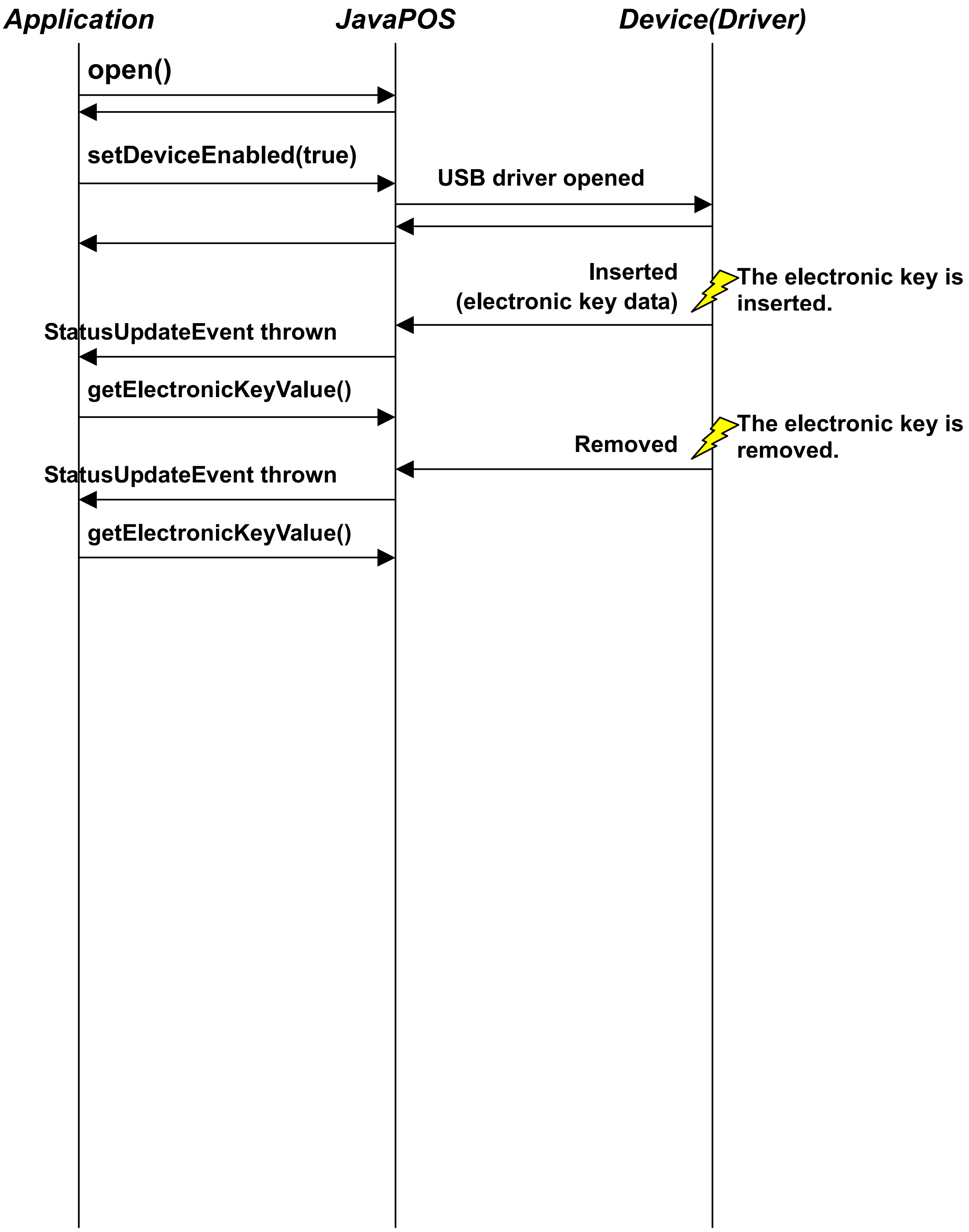
1) Plug-and-Play for USB Cable in Operation

Not supporting the power notification function, this Device does not notify a connection/disconnection status of the USB cable.

1.2.9. Usage Example

This section describes a usage example of this Device Service.

1) Process flow when an electronic key is inserted/removed with the Device enabled



1.2.10. Log

This Device Service outputs a log using Commons Logging.
Depending on a situation, a log at either “INFO", "WARN", or "ERROR" level is output.

1.2.11. Log at INFO Level

A log at INFO level is output to check performance at a start and end of a method and a start and end of a property setting and when an event is thrown.

1.2.12. Log at WARN Level

A log at WARN level is output when an error, which is so minor that there is no need to throw an exception, has occurred. For example, the Device operates using a default value in the module because an unexpected value has been specified to jpos.xml.

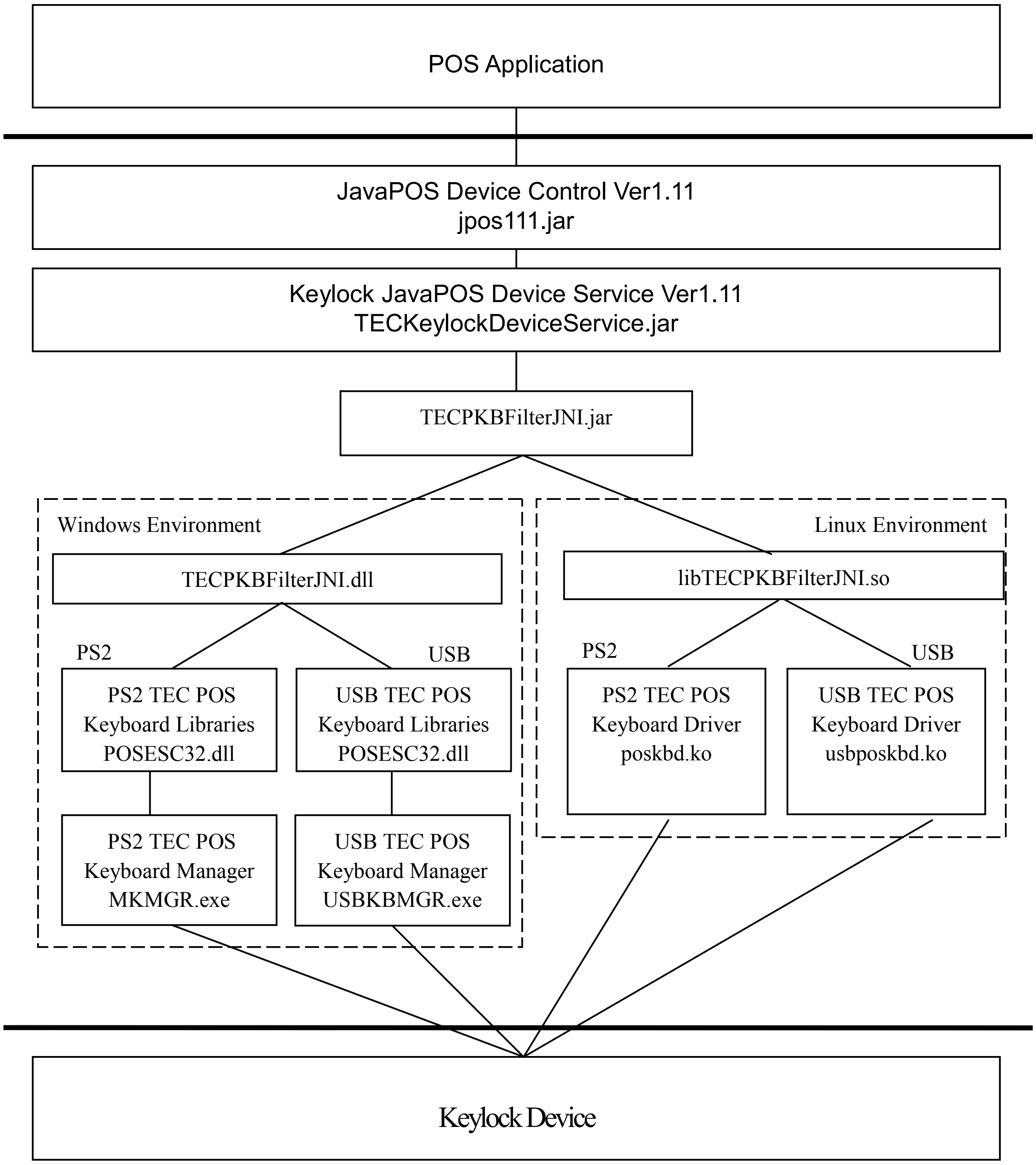
1.2.13. Log at ERROR Level

A log at ERROR level is output when a process in operation stops due to an error. Usually, this type of log is output when an exception is thrown.

1.3. KeylockJavaPOS Device ["PKBST-5x-Keylock"]

1.3.1. Architecture Structure

The KeylockJavaPOS Device uses some software to perform functions.
The software components shown below are required to build an execution environment.



1.3.2. Supported Functions

The functions supported/not supported by this Device Service are as follows:

Supported Functions	Unsupported Functions
Keylock	Power status notification Accumulation of statistics Update of statistics Update of firmware Comparison of firmware version Electronic key
Extended Functions	
None	

Table 11 Keylock JavaPOS Device – Functions

1.3.3. Property Specifications

1.3.3.1. Initial Value of PKBST-5x-Keylock Properties (when opening the Service)

Common Property	Value
AutoDisable:	Not applicable
CapCompareFirmwareVersion:	false
CapPowerReporting:	JPOS_PR_NONE
CapStatisticsReporting:	false
CapUpdateFirmware:	false
CapUpdateStatistics:	false
CheckHealthText:	“” (empty string)
Claimed:	false
DataCount:	Not applicable
DataEventEnabled:	Not applicable
DeviceEnabled:	false
FreezeEvents:	false
OutputID:	Not applicable
PowerNotify:	JPOS_PN_DISABLED
PowerState:	JPOS_PS_UNKNOWN
State:	JPOS_S_IDLE
DeviceControlDescription:	“JavaPOS Keylock Device Control”
DeviceControlVersion:	“1011000”
DeviceServiceDescription:	“TEC JavaPOS Keylock Device Service”
DeviceServiceVersion:	“1011XXX” (*1)
PhysicalDeviceDescription:	“PKBST-5x-Keylock Device"
PhysicalDeviceName:	“PKBST-5x”
Specific Property	**Value**
CapKeylockType:	LOCK_KT_STANDARD
ElectronicKeyValue:	byte[0]
KeyPosition:	0
PositionCount:	8

(*1) Build version is indicated as “XXX” because this manual may not be revised as soon as the module is updated.

Table 12 Keylock JavaPOS Device – Property Value List

1.3.3.2. Details of Properties

1.3.4.2.1 [Common Properties]

CapCompareFirmwareVersion Property

Type

boolean CapCompareFirmwareVersion;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the function to compare the version of a firmware file against the firmware version of the Physical Device is provided.

Value	Meaning
true	This function is supported.
false	This function is not supported.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapPowerReporting Property

Type

int CapPowerReporting;

Mutability

Read Only

Remarks

Holds the power reporting capability.

Value	Meaning
JPOS_PR_NONE	Power reporting function is not available because the state of the Device cannot be determined.
JPOS_PR_STANDARD	Two types of power state (OFF_OFFLINE, ONLINE) can be determined and reported.
JPOS_PR_ADVANCE	Three types of power state (OFF, OFFLINE, ONLINE) can be determined and reported.

The Device returns JPOS_PR_NONE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapStatisticsReporting Property

Type

boolean CapStatisticsReporting;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the statistics reporting function is provided.

Value	Meaning
true	This function is provided.
false	This function is not provided.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapUpdateFirmware Property

Type

boolean CapUpdateFirmware;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the function to update the firmware file of the Device is provided.

Value	Meaning
true	This function is provided.
false	This function is not provided.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CapUpdateStatistics Property

Type

boolean CapUpdateStatistics;

Mutability

Read Only

Remarks

Holds the status that indicates whether or not the statistics update function by the Device Service or the Device is provided.

Value	Meaning
true	This function is provided.
false	This function is not provided.

The Device returns FALSE.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

CheckHealthText Property

Type

String CheckHealthText;

Mutability

Read Only

Remarks

Holds the result of the most recent call to the **checkHealth** method. For details, refer to the description of the **checkHealth** method in the section, “1.2.4.2 Details of Method”.

When the **checkHealth** method has not been called yet, “” (empty string) is held for this property.

Exception

When this property is accessed, no exceptions are thrown.

Claimed Property

Type

boolean Claimed;

Mutability

Read Only

Remarks

Holds the status whether or not the Device is claimed for exclusive access.

Value	Meaning
true	The Device is claimed for exclusive access.
false	The Device is released.

This property always returns FALSE because the Device does not provide the exclusive access function.

Exception

When this property is accessed, no exceptions are thrown.

DeviceEnabled Property

Type

boolean DeviceEnabled;

Mutability

Read / Write

Remarks

Holds the status whether or not the Device is enabled.

Value	Meaning
true	The Device is enabled (in an operational state).
false	The Device is disabled.

By setting this property to TRUE, the Device is enabled (in an operational state) and to FALSE, the Device is disabled.

This property is initialized to FALSE by the **open** method.

Exception

In case of an error when this property is accessed, a Java exception is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_FAILURE,0	Enabling the Device failed.	Enabling the Device failed. Try again after checking a status of the Device. If the error persists, investigate the error.

FreezeEvents Property

Type

boolean FreezeEvents;

Mutability

Read / Write

Remarks

Holds the status whether or not the event is frozen.

Value	Meaning
true	Events are frozen, which means they are queued and not delivered.
false	Events are delivered.

If TRUE, events are frozen, that is, they are queued and not delivered. By setting this property to FALSE, events are allowed to be delivered (if other event delivery conditions are met). If there are some events which have been held in a **FreezeEvents** state, changing this property to FALSE will allow these events to be delivered (if other event delivery conditions are met).

If an interruption by an event is not desirable, the application can choose whether or not the event is to be frozen.

This property is initialized to FALSE by the **open** method.

Exception

In case of an error when this property is accessed, a Java exception is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.

PowerNotify Property

Type

int PowerNotify;

Mutability

Read / Write

Remarks

Holds the power notification function type.

Value	Meaning
JPOS_PN_DISABLED	Power state is not reported. Neither the **StatusUpdateEvent** event for power notification is delivered nor power state is held in the PowerState property.
JPOS_PN_ENABLED	When the **DeviceEnabled** property is TRUE, the **StatusUpdateEvent** event for power notification is delivered and the power state is held in the PowerState property.

Whenever the **open** method is executed, JPOS_PN_DISABLED is always returned because the Device does not provide the power notification function.

Exception

In case of an error when this property is accessed, a Java exception is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	An invalid parameter value was specified.	Check the specified value and specify a valid one.
	The Device is enabled.	Try again after setting the setDeviceEnable property to FALSE.
	Unsupported function was specified.	This function is not supported.

PowerState Property

Type

int PowerState;

Mutability

Read Only

Remarks

Holds the current power condition.

Value	Meaning
JPOS_PS_UNKNOWN	Power state cannot be determined.
JPOS_PS_ONLINE	The Device power is on and the Device is in ready state.
JPOS_PS_OFF	The Device power is off or the Device is not connected.
JPOS_PS_OFFLINE	The Device power is on, but “not ready state” cannot be returned in response to a request.
JPOS_PS_OFF_OFFLINE	The Device is off or in offline state.

Whenever the **open** method is executed, JPOS_PS_UNKNOWN is always returned because the Device does not provide the power notification function.

Exception

When this property is accessed, no exceptions are thrown.

State Property

Type

int State;

Mutability

Read Only

Remarks

Holds the current state of the JavaPOS Device.

Value	Meaning
JPOS_S_CLOSED	The Device is closed.
JPOS_S_IDLE	The Device is in a normal state and is not busy.
JPOS_S_BUSY	The Device is in a normal state and is busy performing output.
JPOS_S_ERROR	An error has been reported. To resume I/O, the Device must be recovered to a normal state.

Whenever the **open** method is executed, JPOS_ S_IDLE is always returned.

Exception

When this property is accessed, no exceptions are thrown.

DeviceControlDescription Property

Type

String DeviceControlDescription;

Mutability

Read Only

Remarks

Holds the Device Control name.

The Device returns the following character string:

“TEC JavaPOS Keylock Device Service”

This property is always readable.

Exception

When this property is accessed, no exceptions are thrown.

DeviceControlVersion Property

Type

String DeviceControlVersion();

Mutability

Read Only

Remarks

Holds the Device Control version number.

The Device returns the following version number:

1011000

This number indicates the Device Control version number is 1.11.0 (in accordance with the JPOS Specification Version 1.11). This property is always readable.

Exception

When this property is accessed, no exceptions are thrown.

DeviceServiceDescription Property

Type

String DeviceServiceDescription;

Mutability

Read Only

Remarks

Holds the Device Service name.

The Device returns the following character string.

“JavaPOS Keylock Device Control”

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

DeviceServiceVersion Property

Type

String DeviceServiceVersion;

Mutability

Read Only

Remarks

Holds the Device Service version number.

The Device returns the following version number.

1011XXX

This number indicates the Device Service version number is 1.11.XXX (in accordance with the JPOS Specification Version 1.11) (The value, “XXX” indicates a build version, which is incremented from 000.). Here, Major no. is 1, Minor number is 11, and Build no. is XXX.

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

PhysicalDeviceDescription Property

Type

String PhysicalDeviceDescription;

Mutability

Read Only

Remarks

Holds the official Physical Device name.

The Device returns the following character string.

“TEC Keylock for PKBST-5x”

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

PhysicalDeviceName Property

Type

String PhysicalDeviceName;

Mutability

Read Only

Remarks

Holds the short Physical Device name.

The Device returns the following character string.

“Keylock for PKBST-5x

This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown..

1.3.4.2.2 [Specific Properties]

CapKeylockType Property

Type

int CapKeylockType;

Mutability

Read Only

Remarks

Holds the value that indicates the keylock type.

Value	Meaning
LOCK_KT_STANDARD	Standard keylock. (Compatible with the JPOS Version 1.10 or earlier)
LOCK_KT_ELECTRONIC	Electronic keylock. Holds the value read from the electronic keylock using the **ElectronicKeyValue** property. Always set to “0” for the **PositionCoun**t and **KeyPositio**n properties.

The Device returns LOCK_KT_STANDARD.

This property is initialized to FALSE by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

ElectronicKeyValue Property

Type

byte[] ElectronicKeyValue;

Mutability

Read Only

Remarks

Holds the value read from the electronic key.

A byte string of 0 size is set because the Device does not support an electronic keylock.

Exception

When this property is accessed, no exceptions are thrown.

KeyPosition Property

Type

int KeyPosition;

Mutability

Read Only

Remarks

Holds the value that indicates the keylock position.

The **StatusUpdateEvent** event is delivered when the keylock position is changed. Accordingly, this property value is updated. This property has one of the following values:

Value	Meaning
LOCK_KP_LOCK	Keylock is in the “locked” position. The value is 1.
LOCK_KP_NORM	Keylock is in the “normal” position. The value is 2.
LOCK_KP_SUPR	Keylock is in the “superviser” position. The value is 3.
Other values	Keylock is in the position other than the above. This value may range from four (4) up to the total number of keylock positions indicated by the **PositionCount** property. For the Device, the range is 4 to 9 (*).

This property is initialized when the open method is called and the Device is disabled.

(*) The Device supports 9 statuses using in total of 8 key positions (status at each of 8 key positions and 1 neutral position between two keys).

Exception

When this property is accessed, no exceptions are thrown.

PositionCount Property

Type

int PositionCount;

Mutability

Read Only

Remarks

Holds the total number of keylock positions that are present on the keylock device.
A value “8” is returned because the Device has in total of 8 key positions.
This property is initialized by the **open** method.

Exception

When this property is accessed, no exceptions are thrown.

1.3.4. Method Specifications

1.3.4.1. Method List

This Device Service provides the following methods.

For the specifications of the checkHealth and directIO methods, refer to the descriptions of these methods in the section, “1.3.4.2 Details of Methods, and for those of other methods, refer to the UPOS Specification.

Common Method	Requirement	Remarks
open	None	
close	open	
claim	open	Not applicable
release	open & claim	Not applicable
checkHealth	open & enable	
compareFirmwareVersion	open & enable	Not supported
directIO	open	Not provided
resetStatistics	open & enable	Not supported
retrieveStatistics	open & enable	Not supported
updateFirmware	open & enable	Not supported
updateStatistics	open & enable	Not supported
Specific Method	**Requirement**	**Remarks**
waitForKeylockChange	open & enable	

Table 13 Keylock JavaPOS Device – Method List

1.3.4.2. Details of Methods

1.3.5.2.1 [Common Methods]

open Method

Type

void open(String logicalDeviceName **) throws JposException;**

Parameter	Description
logicalDeviceName	Specifies a name of the Device to be opened. An attribute value specified by the attribute name “logicalName” in the <JposEntry> element under the <JposEntries> element of the jpos.xml file must be specified to this value.

Remarks

This method is called to open the Device for subsequent I/O.

Exception

When this method is invoked, a **JposException** is thrown.

Exception’s ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_ILLEGAL,0	The Device is open.	The Device has been open.
JPOS_E_NOEXIST,0	The Device was not found.	DeviceService was not found. Check the logicalDeviceName parameter value and the jpos.xml setting.
JPOS_E_NOSERVICE,0	DeviceService is illegal.	DeviceService does not support the methods which it should. Or, the version number is invalid. Check the correct module is used.

close Method

Type

void close() throws JposException;

Remarks

Releases the Device and its resources.

If the **DeviceEnabled** property is TRUE, the Device is disabled.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	–

claim Method

Type

void claim(int timeout **) throws JposException;**

Parameter	Description
timeout	Specifies the maximum wait time in milliseconds to obtain an exclusive access. If "0" is set and the exclusive access is obtained, the method immediately returns the control to the application. For other cases, a JposException is thrown. If JPOS_FOREVER (-1), this method waits as long as needed until the exclusive access is obtained.

Remarks

This method is called when an exclusive access to the Device is requested.

For this Device, a JposException is always thrown because it cannot obatin the exclusive access.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This Device Service does not support this function.

release Method

Type

void release () throws JPOSException;

Remarks

This method is called to release an exclusive access to the Device.

For this Device, a JposException is always thrown because it cannot obatin the exclusive access.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

checkHealth Method

Type

void checkHealth(int level **) throws JposException;**

Parameter	Description
level	Specifies the type of health check to be performed on the Device.

Any of the following values can be specified for the level parameter.

Value	Meaning
JPOS_CH_INTERNAL	Internal test. This parameter is not supported.
JPOS_CH_EXTERNAL	Thorough test. This parameter is not supported.
JPOS_CH_INTERACTIVE	Performs an interactive test of the Device.

Remarks

This Device Service supports a health check of the Interactive level. Regardless of level, the following exceptions are thrown:

Exception’s ErrorCode	CheckHealthText Property	Meaning
JPOS_E_CLOSED	No change	The Device has been closed.
JPOS_E_DISABLED	“HCheck:Disabled”	The Device has been disabled.
JPOS_E_ILLEGAL	“HCheck:Illegal”	Illegal level parameter
JPOS_E_FAILURE	“HCheck:failure”	Captures an exception other than JavaPOS.

1) Internal Level (level=JPOS_CH_INTERNAL)

This Device Service does not support the checkHealth method at an Internal level.
The checkHealth method at an Internal level throws the following exception.

Exception’s ErrorCode	CheckHealthText Property	Meaning
JPOS_E_ILLEGAL	“Internal HCheck:Illegal”	Not supported

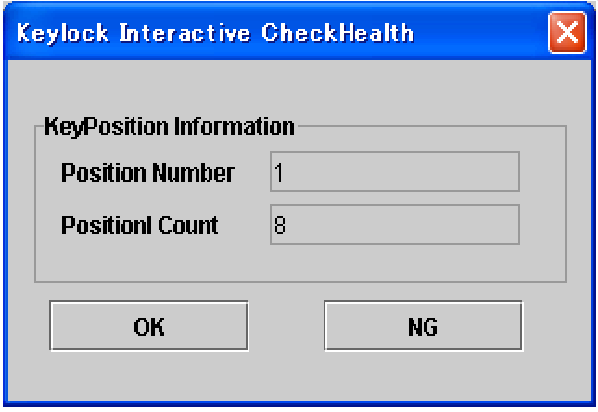
2) External Level (level=JPOS_CH_EXTERNAL)

This Device Service does not support the checkHealth method at an External level.
The checkHealth method at an External level throws the following exception.

Exception’s ErrorCode	CheckHealthText Property	Meaning
JPOS_E_ILLEGAL	“External HCheck:Illegal”	Not supported

3) Interactive Level (level=JPOS_CH_INTERACTIVE)

When the checkHealth method at an Interactive level is invoked, the following dialog box is displayed.



Position Number 1

Positionl Count 8

When the key position is changed after the dialog box appears, a value, which corresponds to the key position, is displayed in the “Position Number” box.
The relationship between the key positions and the position numbers is as shown below.

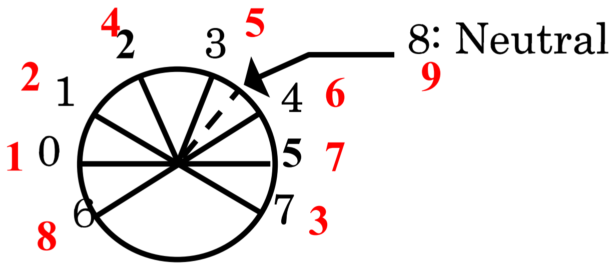


Figure: Relationship between Key Positions and Position Numbers
(Key position (black), Position number (red))

After visually checking the values, press the “OK” button if the values are correct, and “NG” if there is any error.
By pressing the “OK” or “NG” button, the dialog box is closed and the control is returned to the application.
The checkHealth method at an Interactive level throws the following exceptions.

Exception’s ErrorCode	CheckHealthText Property	Meaning
– (No exceptions are thrown.)	“Interactive HCheck:Successful”	Completed with the “OK “ button
JPOS_E_FAILURE	“Interactive HCheck:Error”	Completed with the “NG “ button

Exception

When this method is invoked, a **JposException** is thrown.

Exception’s ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_DISABLED,0	The Device is disabled.	Try again after setting the DeviceEnable property to TRUE.
JPOS_E_ILLEGAL,0	An invalid parameter value was specified.	Check the specified value and specify a valid one.
	Unsupported function was specified.	This function is not supported.
JPOS_E_FAILURE,0	A health check failed.	The health check failed as a result of visual check or an internal error. Try again after checking the Device. If the error persists, investigate the error.

directIO Method

Type

void directIO(int command, **int[]** data, **Object** object **) throws JposException;**

Parameter	Meaning
command	Specifies a command number.
data	An array of one mutable integer of which specific values vary.
object	Specifies an additional data.

The command, data and object parameter values vary depending on the Device Service.

Remarks

This method provides the Device functions, which are not supported by the method and properties, to the application.

For this Device, an exception is alwasy thrown because there are no functions which this method can provide.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

compareFirmwareVersion Method

Type

void compareFirmwareVersion(String firmwareFileName**, int** result**) throws JPOSException;**

Parameter	Meaning
firmwareFileName	Specifies a name of the file containing the firmware or a file containing a set of firmware files.
result	Location in which to return the result of the comparison.

Remarks

The Device does not support this function.
An exception is thrown when this method is called.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

resetStatistics Method

Type

void resetStatistics(String statisticsBuffer**) throws JPOSException;**

Parameter	Meaning
statisticsBuffer	The data buffer defining the statistics that are to be reset.

Remarks

The Device does not support this function.

An exception is thrown when this method is called.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

retrieveStatistics Method

Type

void retrieveStatistics (String statisticsBuffer) throws JPOSException;

Parameter	Meaning
statisticsBuffer	The data buffer defining the statistics to be retrieved and in which the retrieved statistics are placed.

Remarks

The Device does not support this function.

An exception is thrown when this method is called.

Exception

When this method is invoked, a **JposException** is thrown.

Exception's ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

updateFirmware Method

Type

void updateFirmware(String firmwareFileName**) throws JPOSException;**

Parameter	Meaning
firmwareFileName	Specifies either the name of the file containing the firmware or a file containing a set of firmware files that are to be downloaded into the device.

Remarks

The Device does not support this function.

An exception is always thrown whenever this method is called.

Exception

When this method is invoked, a **JposException** is thrown.

ErrorCode & ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

updateStatistics Method

Type

void updateStatistics(String statisticsBuffer**) throws JPOSException;**

Parameter	Meaning
statisticsBuffer	The data buffer defining the statistics with values that are to be updated.

Remarks

The Device does not support this function.

An exception is always thrown whenever this method is called.

Exception

When this method is invoked, a **JposException** is thrown.

ErrorCode & ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_ILLEGAL,0	Unsupported function was specified.	This function is not supported.

1.3.5.2.2 [Specific Methods]

waitForKeylockChange Method

Type

void waitForKeylockChange (int keyPosition**, int** timeout**) throws JposException;**

Parameter	Meaning
keyPosition	Requested keylock position.
timeout	Maximum number of milliseconds to wait for the keylock before returning control back to the application. Any of the following values can be set: -1, 0, and positive value If –1, timeout does not occur.

The keyPosition parameter has one of the following values:

Value	Description
LOCK_KP_ANY	Wait for any keylock position change. Value is 0.
LOCK_KP_LOCK	Wait for keylock position to be set to the “locked” position. Value is 1.
LOCK_KP_NORM	Wait for keylock position to be set to the “normal” position. Value is 2.
LOCK_KP_SUPR	Wait for keylock position to be set to the “supervisor” position. Value is 3.
Other values	Wait for keylock position to be set to one of the positions other than the above. This value ranges from 4 to 9.

Remarks

Waits for a specified keylock position to be set.

Exception

When this method is invoked, a **JposException** is thrown.

Exception’s ErrorCode, ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_DISABLED,0	The Device is disabled.	Try again after setting the setDeviceEnable property to TRUE.
JPOS_E_ILLEGAL,0	An invalid parameter value was specified.	Check the specified value and specify a valid one.
JPOS_E_TIMEOUT,0	A specified timeout period expired.	The timeout period expired before the requested keylock positioning occurred.

1.3.5. Event Specifications

This Device Service throws the following events.
For the event specifications, refer to the UPOS Specification.

1.3.5.1. Event List

Event	Remarks
DataEvent	Not applicable
DirectIOEvent	Not supported
ErrorEvent	Not applicable
OutputCompleteEvent	Not applicable
StatusUpdateEvent	Notifies the application when the Device's operation status changes.

Table 14 Keylock JavaPOS Device – Event List

1.3.5.2. Details of Events

StatusUpdateEvent

Syntax: StatusUpdateEvent(int Status);

This Device Service throws the following StatusUpdateEvent events.

Status Parameter	Meaning
LOCK_KP_LOCK	Keylock is in the "locked" position. Value is 1.
LOCK_KP_NORM	Keylock is in the "normal" position. Value is 2.
LOCK_KP_SUPR	Keylock is in the "supervisor" position. Value is 3.
Other values	Keylock is in one of the positions other than the above. This value ranges from 4 to 9.

DirectIOEvent

An event as a result of the DirectIO method
The Device Service does not throw this type of event.

ErrorEvent

An ErrorEvent event is thrown when a Control's **state** changes to an error state.
The Device Service does not throw this type of event.

DataEvent

A DataEvent event is thrown to indicate data is input from the Device to the application.
The Device Service does not throw this type of event.

OutputCompleteEvent

An OutputCompleteEvent event is thrown when a previously requested asynchronous output has successfully completed.
The Device Service does not throw this type of event.

1.3.6. Setting Information

Setting information of this Device Service is set in the XML file called "jpos.xml".
The <prop> tag in the XML file is a setting item specific to this Device. For details of other tags, <creation>, <vendor>, <jpos>, and <product>, refer to the UPOS Specification.

```
<JposEntries>
    <JposEntry logicalName="TECKeylock_PKBST5x">
        <creation factoryClass="jpos.toshibatec.loader.JavaPOSServiceFactory"
            serviceClass="jpos.toshibatec.services.KeylockService"/>
        <vendor name="TOSHIBA TEC Corporation" url="http://www.toshibatec.co.jp"/>
        <jpos category="Keylock" version="1.11"/>
        <product description="TEC Standard Keylock"
            name="TECKeylock" url="http://www.toshibatec.co.jp"/>

        <prop name="KeylockType" type="String" value="standard"/>
        <prop name="modelName" type="String" value="PKBST-5x"/>
        <prop name="deviceBus" type="String" value="PS2"/>
        <prop name="MechKeyPos0" type="String" value="1"/>
        <prop name="MechKeyPos1" type="String" value="2"/>
        <prop name="MechKeyPos2" type="String" value="4"/>
        <prop name="MechKeyPos3" type="String" value="5"/>
        <prop name="MechKeyPos4" type="String" value="6"/>
        <prop name="MechKeyPos5" type="String" value="7"/>
        <prop name="MechKeyPos6" type="String" value="8"/>
        <prop name="MechKeyPos7" type="String" value="3"/>
        <prop name="MechKeyPos8" type="String" value="9"/>
    </JposEntry>
```

Item Name (jposEntry)	Value
logicalName	Logical name which the application uses. Please specify a logical name which the application uses.

Item Name (Product)	Value
name	TECKeylock (Fixed)

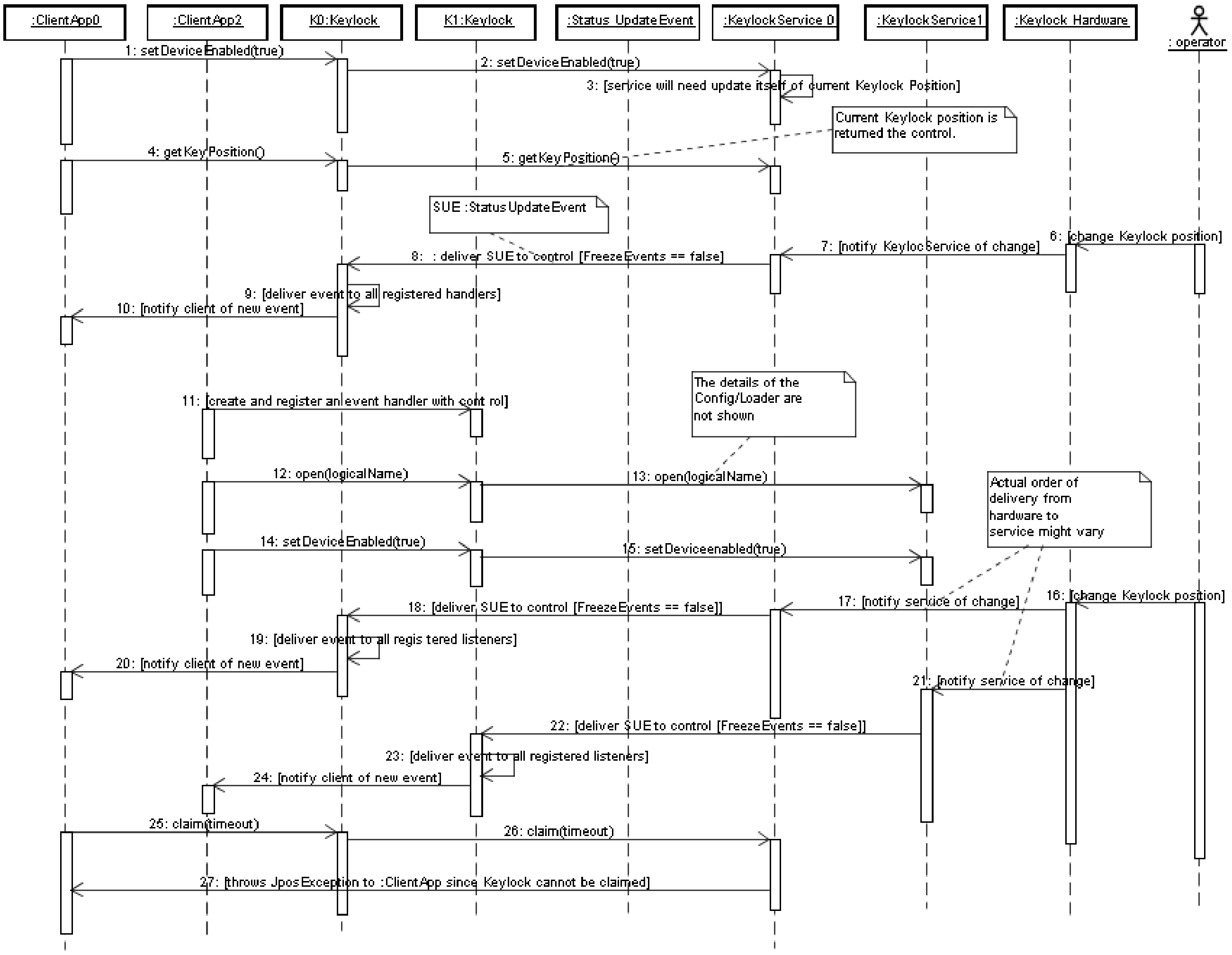
Item Name (prop name)	Value
KeylockType	Specify the Keylock type. Specify “Standard” for the PKBST-5x.
modelName	Specify the model number of the keylock device. Specify “PKBST-5x” for PKBST-50. Specify ”LKBST-65” for LKBST-65. Specify “PKBST-52” for PKBST-52.
deviceBus	Specify the device bus. Specify ”PS2” for the PKBST-50 or LKBST-65. Specify ”USB” for the PKBST-52.
MechKeyPos0	Specify the logical key position number for the physical key position number. Specify ”1” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”5” for the Windows PKBST-52.
MechKeyPos1	Specify the logical key position number for the physical key position number. Specify ”2” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”4” for the Windows PKBST-52.
MechKeyPos2	Specify the logical key position number for the physical key position number. Specify ”4” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”2” for the Windows PKBST-52.
MechKeyPos3	Specify the logical key position number for the physical key position number. Specify ”5” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”1” for the Windows PKBST-52.
MechKeyPos4	Specify the logical key position number for the physical key position number. Specify ”6” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”8” for the Windows PKBST-52.
MechKeyPos5	Specify the logical key position number for the physical key position number. Specify ”7” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”3” for the Windows PKBST-52.
MechKeyPos6	Specify the logical key position number for the physical key position number.

	Specify ”8” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”7” for the Windows PKBST-52.
MechKeyPos7	Specify the logical key position number for the physical key position number. Specify ”3” for the PKBST-50, LKBST-65 and Linux PKBST-52. Specify ”6” for the Windows PKBST-52.
MechKeyPos8	Specify the logical key position number for the physical key position number. Specify ”9” for the PKBST-50, LKBST-65 and PKBST-52.

Table 15 Keylock JavaPOS Device – Setting Information List

1.3.7. Usage Example

This section describes a usage example of this Device Service.
The sequence below assumes that ClientApp0 has already opened the Control.


:ClientApp0
:ClientApp2
K0:Keylock
K1:Keylock
:Status Update Event
:KeylockService 0
:KeylockService1
:Keylock Hardware
: operator
1: setDeviceEnabled(true)
2: setDeviceEnabled(true)
3: [service will need update itself of current Keylock Position]
Current Keylock position is returned the control.
4: getKeyPosition()
5: getKeyPosition()
SUE :Status Update Event
6: [change Keylock position]
7: [notify KeylocService of change]
8: : deliver SUE to control [FreezeEvents == false]
9: [deliver event to all registered handlers]
10: [notify client of new event]
The details of the Config/Loader are not shown
11: [create and register an event handler with cont rol]
12: open(logicalName)
13: open(logicalName)
Actual order of delivery from hardware to service might vary
14: setDeviceEnabled(true)
15: setDeviceenabled(true)
16: [change Keylock position]
17: [notify service of change]
18: [deliver SUE to control [FreezeEvents == false]]
19: [deliver event to all regis tered listeners]
20: [notify client of new event]
21: [notify service of change]
22: [deliver SUE to control [FreezeEvents == false]]
23: [deliver event to all registered listeners]
24: [notify client of new event]
25: claim(timeout)
26: claim(timeout)
27: [throws JposException to :ClientApp since Keylock cannot be claimed]