

TOSHIBA

TOSHIBA POS Terminal

ST-A10/ST-A20

Scanner Application Programmer's Manual
Toshiba TEC JavaPOS

First Edition: July 18, 2008

TOSHIBA TEC CORPORATION

Table of Contents

1. Scanner	1
1.1 Scanner JavaPOS Service [“TECRSScanner”]	1
1.1.1 Supported Operating Systems	1
1.1.2 Supported JavaVM.....	1
1.1.3 Supported Device.....	1
1.1.4 Architecture Overview	1
1.1.5 Property Specifications.....	2
1.1.6 Method Specifications	2
1.1.7 Exception Specifications	3
1.1.7.1 JposException Specifications.....	3
1.1.8 Log.....	3
1.1.8.1 Log at INFO Level	3
1.1.8.2 Log at WARN Level.....	3
1.1.8.3 Log at ERROR Level.....	3
1.2 TEC Scanner JavaPOS Device [“HS530RS”].....	4
1.2.1 Supported Device.....	4
1.2.2 Architecture Structure	4
1.2.3 Supported Functions	4
1.2.4 Property Specifications.....	5
1.2.4.1 Initial Value of HS530RSScanner Properties (when opening the Service).....	5
1.2.4.2 Details of Properties	5
1.2.5 Method Specifications	14
1.2.5.1 Supported/Unsupported Method List	14
1.2.5.2 Details of Methods.....	14
1.2.6 Event Specifications	21
1.2.6.1 Event List.....	21
1.2.6.2 Details of Events	21
1.2.7 Setting Information	23
1.2.8 Usage Example	24
1.2.9 Limitations and Precautions	25

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.

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

Introduction

The Line Display Application User Manual (hereinafter referred to as “this manual”) was documented in accordance with the “UnifiedPOS Specifications Version 1.11” (hereinafter referred 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.
-

1. Scanner

1.1 Scanner JavaPOS Service [“TECScanner”]

A package name of this Device Service is as follows:

“jpos.toshibatec.scanner”

1.1.1 Supported Operating Systems

This Device Service supports the following operating systems.

- Windows 2000
- Windows XP
- Windows Vista
- SUSE Linux Enterprise Desktop 10

1.1.2 Supported JavaVM

This Device Service runs on the following JavaVM.

- Java Runtime Environment 1.4.2

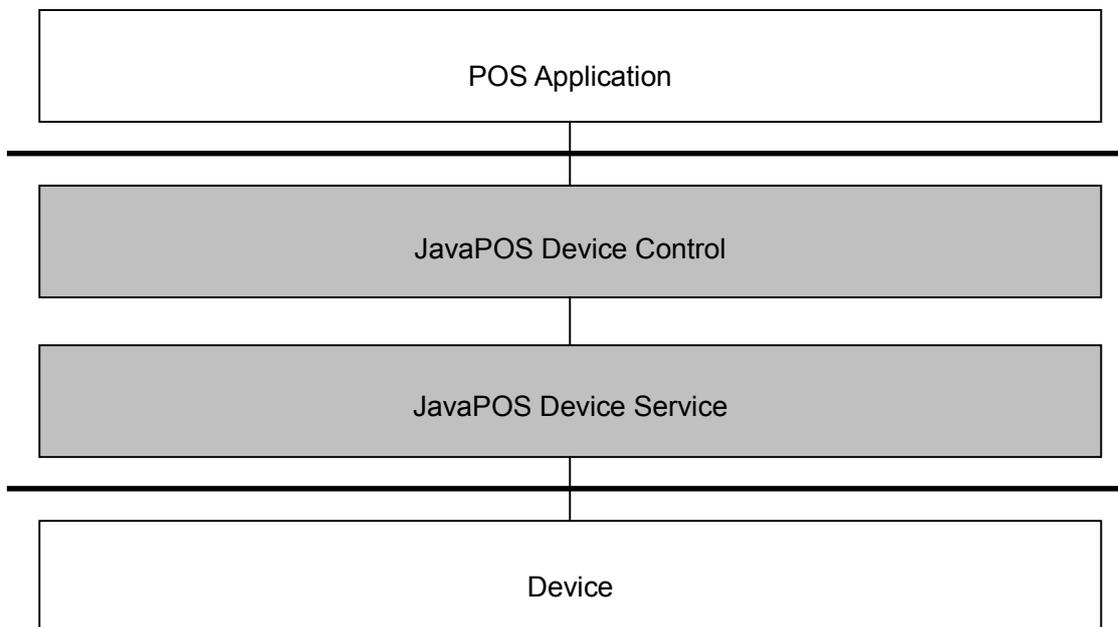
1.1.3 Supported Device

This Device Service supports the following device.

- HS-530-RS Serial Scanner

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 Specifications

The Scanner Device properties in accordance with the UPOS Specification are as follows.

Common Property	Enable Condition	Description
AutoDisable	open	Data-event-driven function to automatically disable the Device
CapCompareFirmwareVersion	open	Support/not support of the function to compare firmware version number
CapPowerReporting	open	Support/not support of the power reporting function
CapStatisticsReporting	open	Support/not support of the statistics reporting function
CapUpdateFirmware	open	Support/not support of the firmware update function
CapUpdateStatistics	open	Support/not support of the statistics update function
CheckHealthText	open	Result of the most recent call to the checkHealth method
Claimed	open	Status of being claimed or released
DataCount	open	Number of obtained data after opening the device
DataEventEnabled	open	Enable/disable of delivery of DataEvent
DeviceEnabled	open & claim	Enable/disable of Device
FreezeEvents	open	Support/not support of the function to temporarily stop event generation
OutputID	Not supported	Holding of the most recent output ID of output device
PowerNotify	open	Enable/disable of the power notification function of the Device
PowerState	open	Current power condition of the Device
State	None	Current state of the Device
DeviceControlDescription	None	Holding of Device Control strings
DeviceControlVersion	None	Holding of Device Control version number
DeviceServiceDescription	open	Holding of Device Service strings
DeviceServiceVersion	open	Holding of Device Service version number
PhysicalDeviceDescription	open	Holding of Physical Device strings
PhysicalDeviceName	open	Holding of Physical Device name

Table 1 JavaPOS Device – Property List

Specific Property	Enable Condition	Description
DecodeData	open	Whether or not data is to be decoded
ScanData	open	Data read from the scanner
ScanDataLabel	open	Bar code data
ScanDataType	open	Bar code type

Table 2 Scanner JavaPOS Device – Property List

1.1.6 Method Specifications

The Scanner Device methods are as follows.

Common Method	Requirement	Remarks
open	-	
close	open	
claim	open	
release	open & claim	
checkHealth	open & claim & enable	
clearInput	open & claim	
clearInputProperties	open & claim	
clearOutput	-	
directIO	open	
compareFirmwareVersion	open & claim & enable	
resetStatistics	open & claim & enable	
retrieveStatistics	open & claim & enable	
updateFirmware	open & claim & enable	
updateStatistics	open & claim & enable	

Table 3 Scanner JavaPOS Device – Method List

1.1.7 Exception Specifications

This Device Service throws the following exceptions.

For details of exception specifications, refer to the UPOS Specification.

Exception	Remarks
JposException	Exceptions are thrown when method execution or a property setting failed, or a process error occurred.

Table 4 Scanner JavaPOS Device – Exception List

1.1.7.1 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.

1.1.8.1 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.

1.1.8.2 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.

1.1.8.3 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 TEC Scanner JavaPOS Device [“HS530RS”]

1.2.1 Supported Device

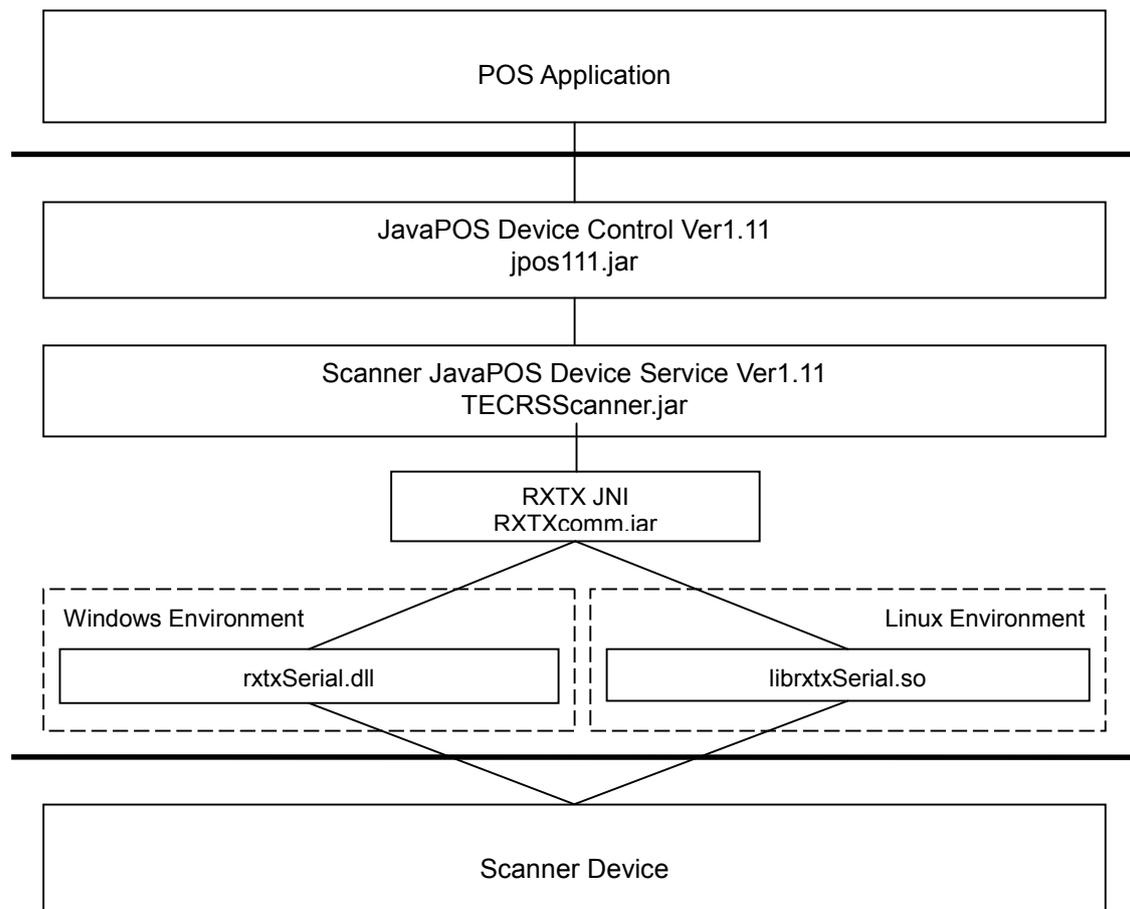
The HS530RS of this Device Service supports the following devices provided by Toshiba TEC.

- Scanner device attached to the ST-A10
HS530RS
- Scanner device attached to the ST-A20
HS530RS

1.2.2 Architecture Structure

The Scanner JavaPOS Device uses some software to perform functions.

The software components shown below are required to build an execution environment.



1.2.3 Supported Functions

The HS530RS functions, supported/not supported by this Device Service, are as follows:

Function	UPOS Ver.	Common / Device	Supported or Not
Power status notification	1.3	Common	Not supported
Accumulation of statistics	1.8	Common	Not supported
Update of statistics	1.8	Common	Not supported
Update of firmware	1.9	Common	Not supported
Comparison of firmware version	1.9	Common	Not supported

Table 5 Scanner JavaPOS Device – Functions

1.2.4 Property Specifications

1.2.4.1 Initial Value of HS530RSScanner Properties (when opening the Service)

Common Property	Value
AutoDisable	false
CapCompareFirmwareVersion	false
CapPowerReporting	JPOS_PR_NONE
CapStatisticsReporting	false
CapUpdateFirmware	false
CapUpdateStatistics	false
CheckHealthText	""
Claimed	false
DataCount	0
DataEventEnabled	false
DeviceEnabled	false
FreezeEvents	false
OutputID	-
PowerNotify	JPOS_PN_DISABLED
PowerState	JPOS_PS_UNKNOWN
State	JPOS_S_IDLE
DeviceControlDescription	"JavaPOS Scanner Device Control"
DeviceControlVersion	"1011000"
DeviceServiceDescription	"TEC JavaPOS Serial Scanner Device Service"
DeviceServiceVersion	"1011XXX" (*1)
PhysicalDeviceDescription	"TEC Serial Scanner Device"
PhysicalDeviceName	"TEC HS-530-RS" (*2)
Specific Property	Value
DecodeData	false
ScanData	byte[0]
ScanDataLabel	byte[0]
ScanDataType	SCAN_SDT_UNKNOWN

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

(*2) Depending on the descriptions of the XML file, the Device's module name is obtained and displayed.

Table 6 Scanner JavaPOS Device – Property Initial Value List

1.2.4.2 Details of Properties

This section details the properties supported by the HS530RS.

1.2.4.2.1 Common Properties

AutoDisable

Type

boolean AutoDisable;

Mutability

Read / Write

Remarks

If TRUE, **DeviceEnabled** property is set to FALSE when **DataEvent** is notified.

If FALSE, **DeviceEnabled** property is not automatically changed even when **DataEvent** is notified.

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

Exception

In case of an error when this property is accessed, a `JposException` is thrown.

Error Code & Error Code Extended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.

CapCompareFirmwareVersion**Type****boolean CapCompareFirmwareVersion;****Mutability****Read Only****Remarks**

Always set to FALSE because this function is not supported by the Device.
Usually, this property is TRUE when the Service/Device provides the function to compare the version of the firmware and the firmware can be updated.

Exception

In case of an error when this property is accessed, 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.

CapPowerReporting**Type****boolean CapPowerReporting;****Mutability****Read Only****Remarks**

Always set to JPOS_PR_NONE because this function is not supported by the Device.

Exception

In case of an error when this property is accessed, 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.

CapStatisticsReporting**Type****boolean CapStatisticsReporting;****Mutability****Read Only****Remarks**

Always set to FALSE because this function is not supported by the Device.
UPOS sets this property to TRUE when the function to report various statistics such as product life is supported.

Exception

In case of an error when this property is accessed, 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.

CapUpdateFirmware**Type****boolean CapUpdateFirmware;****Mutability****Read Only****Remarks**

Always set to FALSE because this function is not supported by the Device.
UPOS sets this property to TRUE when the function to update a firmware via the UPOS is supported.

Exception

In case of an error when this property is accessed, 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.

CapUpdateStatistics**Type**

boolean CapUpdateStatistics;

Mutability

Read Only

Remarks

Always set to FALSE because this function is not supported by the Device.

UPOS sets this property to TRUE when the function to collect statistics is provided and the statistics can be reset to zero.

Exception

In case of an error when this property is accessed, 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.

CheckHealthText**Type**

String CheckHealthText;

Mutability

Read Only

Remarks

Holds the result of the most recent call to the CheckHealth method.

A CheckHealth property value is initialized to an empty string by the open method.

Even once CheckHealth is executed, "Interactive Health Check : Finish" is stored.

Exception

In case of an error when this property is accessed, 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.

Claimed**Type**

boolean Claimed;

Mutability

Read Only

Remarks

If TRUE, an exclusive access to the Device has been obtained.

If FALSE, the Device is released for sharing with other applications.

In many cases, an access to methods and properties are permitted and events can be thrown after an exclusive access to the Device is obtained.

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

Exception

In case of an error when this property is accessed, 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.

DataCount**Type****int DataCount;****Mutability****Read Only****Remarks**

Number of enqueued DataEvents at the control is stored. This property is initialized to "0" by the **open** method.

Exception

In case of an error when this property is accessed, 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.

DataEventEnabled**Type****boolean DataEventEnabled;****Mutability****Read / Write****Remarks**

If TRUE, a **DataEvent** is delivered as soon as input data is enqueued. If changed to TRUE and some input data is already queued, a **DataEvent** is delivered immediately. (Note that other conditions may delay "immediate" delivery: If **FreezeEvents** is TRUE or other event is already being processed by the application, **DataEvent** will remain enqueued at the JPOS Service until the status changes.)

If FALSE, input data is queued for later delivery to the application. Also, if an input error occurs, **ErrorEvent** is not delivered while **DataEventEnabled** is FALSE.

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

Exception

In case of an error when this property is accessed, 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.

DeviceEnabled**Type****boolean DeviceEnabled;****Mutability****Read / Write****Remarks**

If TRUE, the Device is in an enabled state (in an operational state). Whenever changed to TRUE, the Device is enabled.

If FALSE, the Device is in a disabled state. Whenever changed to FALSE, the Device is disabled.

Before using the Device, the application must set this property to TRUE.

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

* This device can't confirm connection status while Host and Device, Therefore Device Enabled process is finished normally if service can get the right to use to a communication port.

Exception

In case of an error when this property is accessed, 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_NOTCLAIMED,0	The Device is not claimed.	Try again after executing the Claim 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**Type****boolean FreezeEvents;****Mutability****Read / Write****Remarks**

If TRUE, the Control does not deliver events.

The Control holds the events until the FreezeEvents state is cleared.

If FALSE, the Control delivers events. 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 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 JposException is thrown.

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

OutputID**Type****int OutputID;****Mutability****Read Only****Remarks**

Always set to "0" because this function is not supported by the Device.

UPOS holds the identifier of the most recently started asynchronous output.

Exception

In case of an error when this property is accessed, 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.

PowerNotify**Type****int PowerNotify;****Mutability****Read / Write****Remarks**

Always set to JPOS_PN_DISABLED because this property is not supported by the Device.

Exception

In case of an error when this property is accessed, 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	An invalid parameter value was specified.	Check the specified value and specify a valid one.
	The Device is enabled.	Try again after executing setDeviceEnabled(false).
	Unsupported function was specified.	This function is not supported.

PowerState**Type**

int PowerState;

Mutability

Read Only

Remarks

Always set to JPOS_PS_UNKNOWN because this property is not supported by the Device.

Exception

In case of an error when this property is accessed, 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	An invalid parameter value was specified.	Check the specified value and specify a valid one.
	The Device is enabled.	Try again after executing setDeviceEnabled(false).
	Unsupported function was specified.	This function is not supported.

State**Type**

int State;

Mutability

Read Only

Remarks

Indicates a current state of the Control.

This property is always readable as long as the Service is loaded.

Value	Meaning
JPOS_S_CLOSED	The Control is closed.
JPOS_S_IDLE	The Control is in a normal state and is not busy.
JPOS_S_ERROR	In an error state. This value is set only by the ErrorEvent event handler.

Exception

If the module has been successfully loaded, no exceptions will be thrown when this property is accessed.

DeviceControlDescription**Type**

String DeviceControlDescription;

Mutability

Read Only

Remarks

This property describes a Device Control class.

This property is always readable as long as the Service is loaded.

"JavaPOS Scanner Device Control" is set to the Device.

Exception

If the module has been successfully loaded, no exceptions will be thrown when this property is accessed.

DeviceControlVersion**Type**

int DeviceControlVersion;

Mutability

Read Only

Remarks

This property indicates the version number of the Device Control class.

This property is always readable as long as the Service is loaded.

The version number of the Device is 1011000, which indicates the Device is in accordance with the JPOS 1.11.000.

Exception

If the module has been successfully loaded, no exceptions will be thrown when this property is accessed.

DeviceServiceDescription**Type**

String DeviceServiceDescription;

Mutability

Read Only

Remarks

This property describes the Device Service class.

"TEC JavaPOS Serial Scanner Device Service" is set for the Device.

Exception

In case of an error when this property is accessed, 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.

DeviceServiceVersion**Type**

int DeviceServiceVersion;

Mutability

Read Only

Remarks

This property indicates the version number of the Device Service class.

The version number of the Device is "1011XXX".

The value, "XXX" indicates a build version, which is incremented from 001.

Exception

In case of an error when this property is accessed, 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.

PhysicalDeviceDescription**Type**

String PhysicalDeviceDescription;

Mutability

Read Only

Remarks

This property describes a Physical Device.
 “TEC Serial Scanner Device” is set for the Device.

Exception

In case of an error when this property is accessed, 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.

PhysicalDeviceName**Type**

String PhysicalDeviceName;

Mutability

Read Only

Remarks

This property describes a name of the Physical Device.
 The name of the Physical Device is “HS530RS”.

Exception

In case of an error when this property is accessed, 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.

1.2.4.2.2 Specific Properties**DecodeData****Type**

boolean DecodeData;

Mutability

Read / Write

Remarks

If TRUE, ScanData is decoded into the ScanDataLabel and ScanDataType properties.
 This property is initialized to FALSE by the **open** method.

Exception

In case of an error when this property is accessed, 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.

ScanData**Type**

byte[] ScanData;

Mutability

Read Only

Remarks

Holds the bar code data read from the scanner.
 This property is set just before DataEvent is delivered.

Exception

In case of an error when this property is accessed, 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.

ScanDataLabel**Type****byte[] ScanDataLabel;****Mutability****Read Only****Remarks**

Holds the decoded bar code data.

This property is set just before DataEvent is delivered.

Exception

In case of an error when this property is accessed, 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.

ScanDataType**Type****int ScanDataType;****Mutability****Read Only****Remarks**

Holds the decoded bar code label type.

When DecodeData is FALSE, SCAN_SDT_UNKNOWN is set to this property.

This property is set just before DataEvent is delivered.

Exception

In case of an error when this property is accessed, 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.

1.2.5 Method Specifications

1.2.5.1 Supported/Unsupported Method List

Supported/unsupported methods by this Device (HS530 RS Scanner) are as follows:

Common Method	Supported or Not
open	Mandatory supported
close	Mandatory supported
claim	Mandatory supported
release	Mandatory supported
checkHealth	Only Interactive Check Health is supported
clearInput	Supported
clearInputProperties	Supported
clearOutput	Not supported
directIO	Not provided
compareFirmwareVersion	Not supported
resetStatistics	Not supported
retrieveStatistics	Not supported
updateFirmware	Not supported
updateStatistics	Not supported

Table 7 Scanner JavaPOS Device(HS530 RS Scanner) – Method List

1.2.5.2 Details of Methods

open Method

Type

open (String *logicalDeviceName*) throws JPOSException;

Remarks

This method is called to open the Device.

The device name specifies the Device which should be used among the Devices supported by this Control class.

The ***logicalDeviceName*** must be the one specified by JPOS.xml.

Exception

In case of an error when this method is invoked, a JposException is thrown.

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 detected.	DeviceService was not found. Check the logicalDeviceName parameter value and jpos.xml setting.
JPOS_E_NOSERVICE,0	DeviceService is invalid.	DeviceService does not support the method which it should. Or, the version number is invalid. Check the correct module is used.

close Method

Type

close () throws JPOSException;

Remarks

This method is called to release the Device and its resources.

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

If the **Claimed** property is TRUE, an excessive access to the Device is released first.

Exception

In case of an error when this method is invoked, a JposException is thrown.

ErrorCode & ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	–

claim Method**Type**

claim (int *Timeout*) throws JPOSException;

Remarks

The *Timeout* parameter indicates the maximum wait time in milliseconds to obtain an exclusive access. If "0", the method immediately returns the result even when the method failed to obtain the exclusive access. If JPOS_FOREVER (-1), this method waits as long as needed until the exclusive access is obtained.

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

The Device cannot be used unless the exclusive access is obtained.

When the exclusive access is successfully obtained, the **Claimed** property is changed to TRUE.

Exception

In case of an error 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_CLAIMED,0	The Device is claimed by other instance.	Try again after other instance releases the Device.
JPOS_E_FAILURE,0	The claim process failed.	Try again. If the error still persists, there may be a communication problem with the Device. For example, the connector is disconnected.

release Method**Type**

release () throws JPOSException;

Remarks

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

If the **DeviceEnabled** property is TRUE and the Device is exclusively used, the Device is disabled.

Exception

In case of an error 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	The Release process failed.	The state is automatically settled so that the Claim method can be executed again. When using the Device, always execute the Claim method.

checkHealth Method**Type**

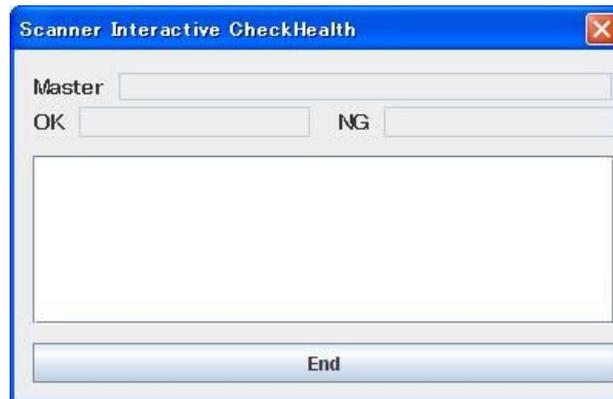
checkHealth (int *Level*) throws JPOSException;

Remarks

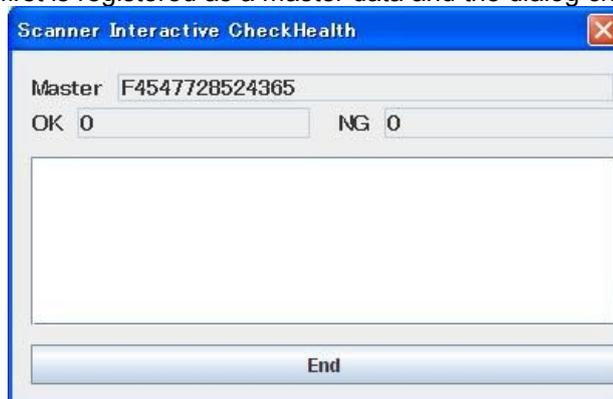
The *Level* parameter indicates the following types of health check to be performed on the Device.

Value	Meaning
JPOS_CH_INTERNAL	Performs a health check that does not physically use the Device. The Device is tested by internal tests to the extent possible. This parameter is not supported.
JPOS_CH_EXTERNAL	Performs a thorough test using the Device. This parameter is not supported.
JPOS_CH_INTERACTIVE	Performs an interactive test with the Device. The supporting Device Service will typically display a modal dialog box to present test options and results.

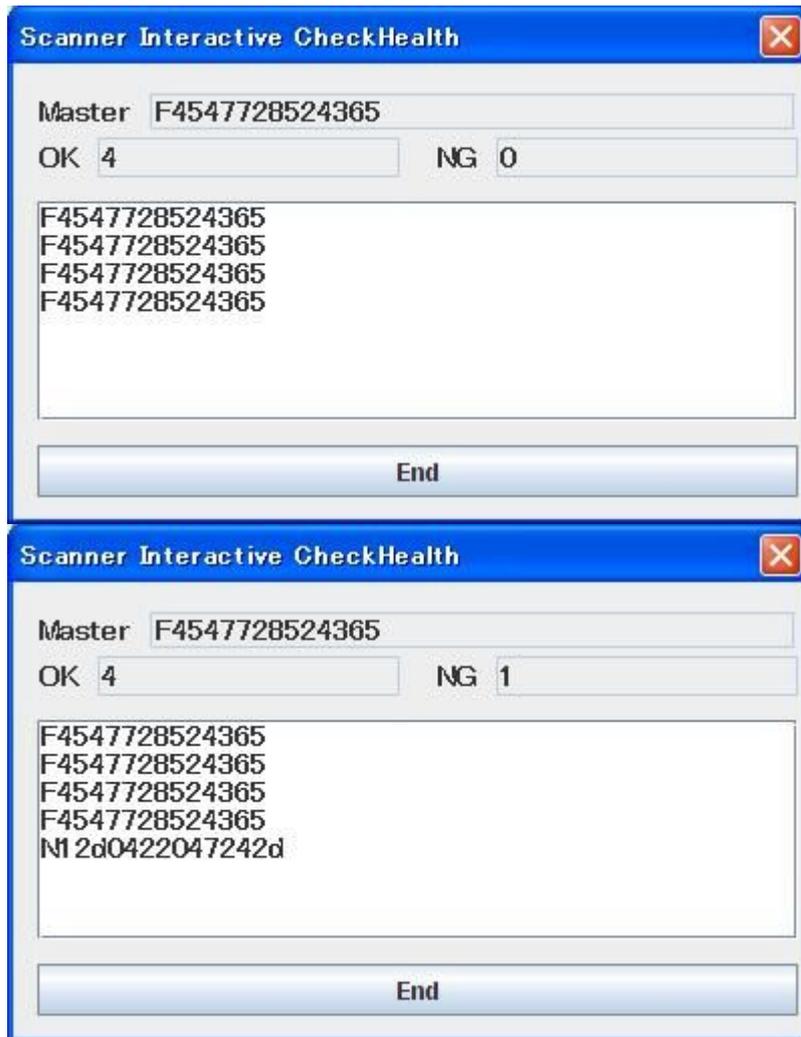
When the checkHealth method is performed at an interactive level, the following dialog box is displayed.



Here, the bar code read first is registered as a master data and the dialog changes as shown below.



Subsequent scanned bar codes are displayed in the center list box and compared with the master data. The number of "OK" and "NG" is incremented.



If the number of "NG" is "1" or larger when the "End" button is pressed, such status is described in CheckHealthText.

Exception

This Device Service only supports a health check at an Interactive level.

Regardless of level, the checkHealth method throws the following exceptions.

ErrorCode & ErrorCodeExtended	Meaning	Error Handling
JPOS_E_CLOSED,0	The Device has been closed.	Try again after executing the open method.
JPOS_E_NOTCLAIMED,0	The Device is not claimed.	Try again after claiming the Device.
JPOS_E_DISABLED,0	The Device has been disabled.	Try again after setting the DeviceEnabled property to TRUE.
JPOS_E_ILLEGAL,0	An invalid level parameter value was specified.	Try again after setting the level parameter to JPOS_CH_INTERACTIVE.
JPOS_E_FAILURE,0	An error occurred during CheckHealth.	Not a JavaPOS exception. Read a message and take necessary action. Try again after checking the Device status. If the error persists, investigate the error.

The checkHealthText property values vary as shown below.

Exception's ErrorCode	CheckHealthText Property
JPOS_E_CLOSED	No change
JPOS_E_NOTCLAIMED	"HCheck:Exclusive"
JPOS_E_DISABLED	"HCheck:Disabled"
JPOS_E_ILLEGAL	"HCheck:Illegal"
JPOS_E_FAILURE	"HCheck:failure"

1) Internal Level (level=JPOS_CH_INTERNAL)

This Device Service does not support the checkHealth method at an Internal level.

The following exception is thrown when the checkHealth method at an Internal level is invoked.

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 following exception is thrown when the checkHealth method at an External level is invoked.

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

3) Interactive Level (level=JPOS_CH_INTERACTIVE)

The following exceptions are thrown when the checkHealth method at an Interactive level is invoked.

Exception's ErrorCode	CheckHealthText Property	Meaning
– (No exceptions are thrown.)	"Interactive HCheck: Successful"	Completed only with "OK".

clearInput Method**Type**

clearInput ()throws JPOSException;

Remarks

Clears **DataEvent** and **ErrorEvent** being buffered. An expression, "being buffered" indicates a state where usually the method is waiting for DataEventEnabled to be set to TRUE and FreezeEvents to be set to FALSE.

Exception

In case of an error 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.

clearInputProperties Method**Type**

clearInputProperties () throws JPOSException;

Remarks

Initializes all properties which are updated by DataEvent or ErrorEvent. Therefore, this method does not clear the DataCount property or the State property.

Exception

In case of an error 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.

ClearOutput Method**Type**

ClearOutput() throws JPOSException;

Remarks

An exception is always thrown because the Device is an input device.
Usually this method is used to clear all buffered output data in the Device.

Exception

In case of an error 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.	The function is not supported.

directIO Method**Type**

directIO (int *Command*, int *pData*, Object *pString*) throws JPOSException;

Remarks

The Device does not support this function.

Exception

This Device Service provides no functions that can be performed using the directIO method.

This directIO method throws the following exceptions regardless of command parameter values.

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.	The function is not supported.

compareFirmwareVersion Method**Type**

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

Remarks

The Device does not support this function.

Exception

In case of an error 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..

resetStatistics Method**Type**

resetStatistics(String *statisticsBuffer*) throws JPOSException;

Remarks

An exception is always thrown when this method is called because the Device does not support this function.

Exception

In case of an error 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.

retreiveStatistics Method**Type**

retreiveStatistics(String *StatisticsBuffer*) throws JPOSException;

Remarks

An exception is always thrown when this method is called because the Device does not support this function.

Exception

In case of an error 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.

updateFirmware Method**Type**

updateFirmware(String firmwareFileName) throws JPOSException;

Remarks

An exception is always thrown when this method is called because the Device does not support this function.

Exception

In case of an error 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**

updateStatistics(String statisticsBuffer) throws JPOSException;

Remarks

An exception is always thrown when this method is called because the Device does not support this function.

Exception

In case of an error 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.

1.2.6 Event Specifications

This Device Service throws the following events.

For details of event specifications, refer to the UPOS Specification.

1.2.6.1 Event List

Common Event	Remarks
DataEvent	Notifies a data input from the Device.
DirectIOEvent	Not Supported
ErrorEvent	The Device Service detects an error and notifies of it.
OutputCompleteEvent	Not Supported
StatusUpdateEvent	Not Supported

Table 8 Scanner JavaPOS Device – Event List

1.2.6.2 Details of Events

DataEvent Event

Type

DataEvent (int Status);

Remarks

An event notification to indicate there is an incoming data from the Device

ErrorEvent Event

This Device Service throws the following ErrorEvent.

Type

**void ErrorEvent (int *ErrorCode*, int *ErrorCodeExtended*,
int *ErrorLocus*, Object *pErrorResponse*);**

Remarks

Parameter	Description
<i>ErrorCode</i>	A code which indicates a cause of the error event. For values, refer to the below.
<i>ErrorCodeExtended</i>	An extended code which indicates a cause of the error event. For values, refer to the below.
<i>ErrorLocus</i>	Location of the error. For values, refer to the below.
<i>pErrorResponse</i>	A pointer to the area which specifies a process to be performed for the error event. For values, refer to the below.

The *ErrorLocus* parameter is one of the following:

Parameter	Description
JPOS_EL_INPUT	An error occurred while processing event-driven input. No previously buffered input data is available.
JPOS_EL_INPUT_DATA	An error occurred while processing event-driven input. Previously queued data is available.

The contents at the location pointed by the *pErrorResponse* parameter are preset to a default value based on *ErrorLocus*. The application may change them to one of the following:

Parameter	Description
JPOS_ER_CLEAR	Clears all buffered input data, and the error state is exited. Default when <i>ErrorLocus</i> is JPOS_EL_INPUT.
JPOS_ER_CONTINUEINPUT	Used only when <i>ErrorLocus</i> is JPOS_EL_INPUT_DATA. Acknowledges the error and directs the Control to continue processing. Although the Control remains in the error state, an additional DataEvent is delivered as directed by the DataEventEnabled property. When all input has been delivered and the DataEventEnabled property is again set to TRUE, another ErrorEvent is delivered with JPOS_EL_INPUT. Default when <i>ErrorLocus</i> is JPOS_EL_INPUT_DATA.

An event is thrown when an error is found while the Device is reading data. To perform correct

application sequence, an input error event is not delivered until the **DataEventEnabled** property is set to TRUE.

Regardless of the **AutoDisable** property value, the Control remains enabled.

StatusUpdateEvent Event

When the Scanner's power state changes, this event is thrown.

However, this event is not thrown from this Device Service which does not detect a power state.

DirectIOEvent Event

An event to the DirectIO method.

However, this event is not thrown from this Device Service which does not provide an access with the DirectIO method.

OutputCompleteEvent Event

An event to report the completion of an asynchronous output request

This event is not thrown from this Device Service which does not perform an asynchronous output process (writing data to track).

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=" defaultScanner ">
  <creation factoryClass="jpos.toshibatec.loader.JavaPOSServiceFactory"
    serviceClass="jpos.toshibatec.services.ScannerService"/>
  <vendor name="TOSHIBA TEC Corporation" url="http://www.toshibatec.co.jp"/>
  <jpos category="Scanner" version="1.11"/>
  <product description=" TEC Serial Scanner"
    name="TECScanner" url="http://www.toshibatec.co.jp"/>

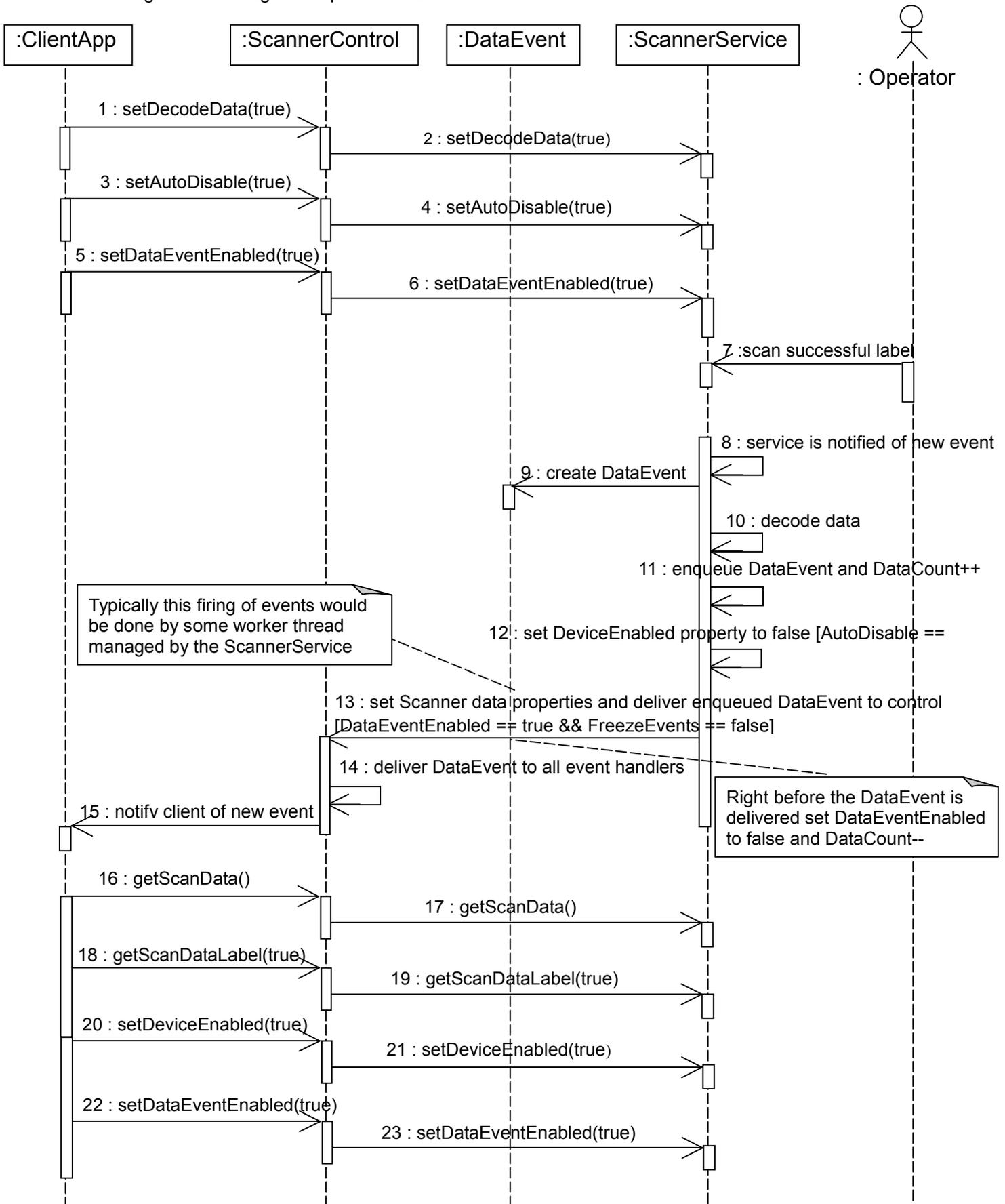
  <prop name="modelName" type="String" value="HS530RS"/>
  <prop name="portName" type="String" value="COM3"/>
  <prop name="startBit" type="Byte" value="2"/>
  <prop name="dataBits" type="String" value="7"/>
  <prop name="parity" type="String" value="Even"/>
  <prop name="flowControl" type="String" value="None"/>
  <prop name="stopBits" type="String" value="2"/>
  <prop name="deviceBus" type="String" value="RS232"/>
  <prop name="endBit" type="Byte" value="3"/>
  <prop name="baudRate" type="String" value="9600
</JposEntry>
```

prop name	Value
logicalName	It is the logical name that an application uses. Mention the logicalname that an application uses.
modelName	Scanner model Name
portName	Serial port name
startBit	Start bit
dataBits	Data bit 7 bits or 8 bits Default: 7 bits
parity	Parity None, odd or even. Default: even
flowControl	Flow control
stopBits	Stop bit 1 bit or 2 bits Default: 2 bits
deviceBus	Method of connection with the Device
endBit	End bit
BaudRate	Baud rate 300, 600, 1200, 2400, 4800, 9600, or 19200 bps Default: 9600

Table 9 Scanner JavaPOS Device –Setting Information List

1.2.8 Usage Example

The following shows a usage example of this Device Service.



1.2.9 Limitations and Precautions

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

1) About the number of the reservation restrictions of DataEvent

In this service, DataEvent is reserved 20 max if DataEventEnabled=false or FreezeEvent=true.

Device cannot read the next barcode till an event is sent to an application by the property operation mentioned above.

2) Relationship between ScanData property and Device

The ScanData property holds the data read from the scanner, excluding header information such as STX character and terminator character such as ETX and CR characters.

All other data is stored in the ScanData property. Some of other data are:

- Symbology character which indicates bar code symbology
- Number of digits of bar code data

The format of ScanData, including the above information, is as follows:

Symbology character	Number of digits	Bar code data
---------------------	------------------	---------------

The symbology characters are used to identify bar code symbologies.

The number of digits is 1 or 2.

Bar Code Symbology	HS530RS
UPC-A	'A'
UPC-E	'E'
EAN-13	'F'
EAN-8	'FF'
UPC-D3	'D3'
2of5 Standard	'H'
2of5 Interleaved	'I'
CODABAR(NW-7)	'N'
CODE39	'M'
CODE93	'L'
CODE128	'K'

Table 10 ScanData Property's Symbology Characters

The number of digits of bar code data is two. For UPC and EAN/JAN bar codes, the number of digits is omitted.

Bar code data is the data of bar code itself.

For the format, refer to the table in the next page, but please note the format changes in accordance with a change in a device setting because the data format is defined by the Device. The format given in the table is used when the Device operates with the default settings.

Bar Code Symbology	Barcode Data
UPC-A	S X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X ₇ X ₈ X ₉ X ₁₀ C/D 12 digits, with C/D
UPC-E	0 X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ 7 digits, without C/D
EAN-13	Y ₁ Y ₂ X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X ₇ X ₈ X ₉ X ₁₀ C/D 13 digits, with C/D
EAN-8	Y ₁ Y ₂ X ₁ X ₂ X ₃ X ₄ X ₅ C/D 8 digits, with C/D
UPC-A with supplemental code	S X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X ₇ X ₈ X ₉ X ₁₀ C/D X _a X _b X _c X _d X _e , or S X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X ₇ X ₈ X ₉ X ₁₀ C/D X _a X _b
UPC-E with supplemental code	0 X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X _a X _b X _c X _d X _e , or 0 X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X _a X _b
EAN-13 with supplemental code	Y ₁ Y ₂ X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X ₇ X ₈ X ₉ X ₁₀ C/D X _a X _b X _c X _d X _e , or Y ₁ Y ₂ X ₁ X ₂ X ₃ X ₄ X ₅ X ₆ X ₇ X ₈ X ₉ X ₁₀ C/D X _a X _b
EAN-8 with supplemental code	Y ₁ Y ₂ X ₁ X ₂ X ₃ X ₄ X ₅ C/D X _a X _b X _c X _d X _e , or Y ₁ Y ₂ X ₁ X ₂ X ₃ X ₄ X ₅ C/D X _a X _b
UPC-Dx	X ₁ X ₂ X ₃ X ₄ X ₅
2of5 Standard	X ₁ X ₂ X ₃ X ₄ X ₅ without start/stop code
2of5 Interleaved	X ₁ X ₂ X ₃ X ₄ X ₅ without start/stop code
CODABAR(NW-7)	A X ₁ X ₂ A with start/stop code (a/b/c/d)
CODE39	X ₁ X ₂ X ₃ X ₄ X ₅ without start/stop code
CODE93	X ₁ X ₂ X ₃ X ₄ X ₅ without start/stop code without FNC code and C/D
CODE128	X ₁ X ₂ X ₃ X ₄ X ₅ without start/stop code without FNC code and C/D

S: Number system character

Y: Flag number character

0: Leading character for converting the number of digits of data to be transmitted

C/D: Check digit

A: Start/stop code

Table 11 Barcode Data Format (Default Value)