

# **USB PKBST MSR OPOS Control Application User Manual**

First Edition: February 05, 2010

**TOSHIBA TEC CORPORATION**

Copyright (C) 2010  
TOSHIBA TEC CORPORATION

---

No. EAA-02151

[illegible]

This specification describes the matters that require attention and the methods Application Programming to utilize TEC OPOS Control.

It is assumed that the reader already possesses some knowledge of.

- General features of the POS peripherals.
- General features of TEC POS Terminal and TEC POS peripherals.
- Terms and Architecture for the OLE Control and OLE Automation.
- The OLE for Retail POS (abbr. OPOS) and the Application Programmer's Guide (abbr. APG).

Copyright © 2010 Toshiba TEC Corporation All rights reserved. It is prohibited to use or duplicate a part or whole of this document without the permission of Toshiba TEC Corporation.

This document is subject to change without prior notice.

#### Trademark Notification

- \* Microsoft, Windows, Windows XP, WEPOS, and POSReady 2009 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".

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

## **Table of Contents**

|   |          |
|---|----------|
| <b>1. Magnetic Stripe Reader .....</b>                          | <b>4</b> |
| 1.1 USB PKBST MSR CONTROL ["MCRSTUB"].....                      | 4        |
| 1.1.1 Applicable Models and Operating Systems .....             | 4        |
| 1.1.2 Software Structure.....                                   | 5        |
| 1.1.3 Functions .....   | 6        |
| 1.1.4 CheckHealth Method Specifications .....                   | 7        |
| 1.1.5 DirectIO Specifications .....                             | 9        |
| 1.1.6 OPOS Registry.....  | 9        |
| 1.1.7 Limitations and Precautions .....                         | 10       |
| 1.1.8 Usage Example .....                                       | 12       |
| 1.1.9 Log.....  | 13       |
| 1.1.10 Result When Property/Method is Executed .....            | 14       |
| <br>Table 1 USB PKBST MSR Control – Functions .....             | <br>6    |
| Table 2 USB PKBST MSR Control – Property Values (in part) ..... | 6        |
| Table 3 USB PKBST MSR Control – Registries .....                | 9        |

# 1. Magnetic Stripe Reader

## 1.1 USB PKBST MSR CONTROL ["MCRSTUB"]

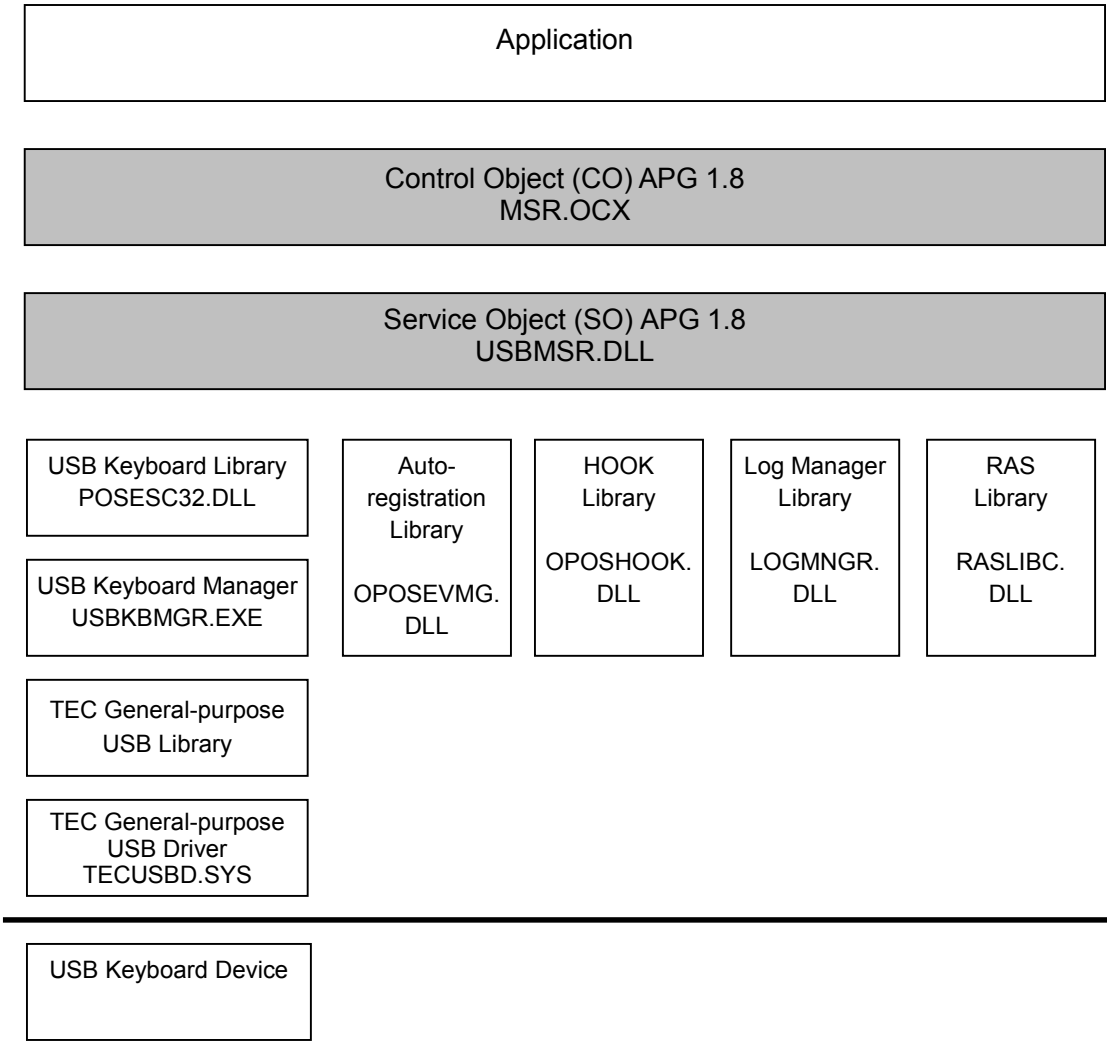
### 1.1.1 Applicable Models and Operating Systems

| Model   | Interface | Device Name (*1) |
|---|-----------|------------------|
| PKBST-52 (USB) MSR  | USB I/F   | "MCRSTUB"        |
| Operating System  |           |                  |
| Windows XP Professional<br>Windows Embedded for Point of Service(WEPOS)<br>Windows Embedded POSReady 2009 |           |                  |

(\*1) Device names are used by the Open method.

1.1.2 Software Structure

The software structure of the USB PKBST MSR Control is as shown below.



 Consist of this control.

### 1.1.3 Functions

| Functions supported                     | Functions not supported  |
|---|--------------------------|
| Power notification                      | Reading of JIS-I track   |
| Error report about card level           | Collection of statistics |
| Error report about track level          | Reset of statistics      |
| Reading of ISO, JIS-II track            | Change of statistics     |
| Reading of start and end sentinels (*1) |                          |

\*1: This function is not supported if "DataType"="MCRST" is described in the registry.

**Table 1 USB PKBST MSR Control – Functions**

The following table shows only the device-dependent properties.

| Common property          | Value                                      |
|--------------------------|--|
| ControlObjectDescription | "TEC OPOS MSR Control Object"              |
| ControlObjectVersion     | "1008XXX" (*2)                             |
| ServiceObjectDescription | "TEC OPOS USB Keyboard MSR Service Object" |
| ServiceObjectVersion     | "1008XXX" (*2)                             |
| DeviceDescription        | "MSR on PKBST-5xU POS Keyboard"            |
| DeviceName               | "MCRST UB"                                 |
| CapPowerReporting        | OPOS_PR_STANDARD                           |
| CapStatisticsReporting   | FALSE                                      |
| CapUpdateStatistics      | FALSE                                      |
| Exclusive property       | Value                                      |
| CapISO                   | TRUE                                       |
| CapJISOne                | FALSE                                      |
| CapJISTwo                | TRUE                                       |
| CapTransmitSentinels     | TRUE (*1)                                  |

(\*1) The value will become FALSE if "DataType"="MCRST" is described in the registry.

(\*2) Build version is indicated as "XXX" because this document may not be revised every time the module is updated.

**Table 2 USB PKBST MSR Control – Property Values (in part)**

### 1.1.4 CheckHealth Method Specifications

#### 1) Internal Level (OPOS\_CH\_INTERNAL)

| Return Value (ResultCode) | CheckHealthText           | Meaning        |
|---------------------------|---------------------------|----------------|
| OPOS_E_ILLEGAL            | "Internal Hcheck:Illegal" | Not supported. |

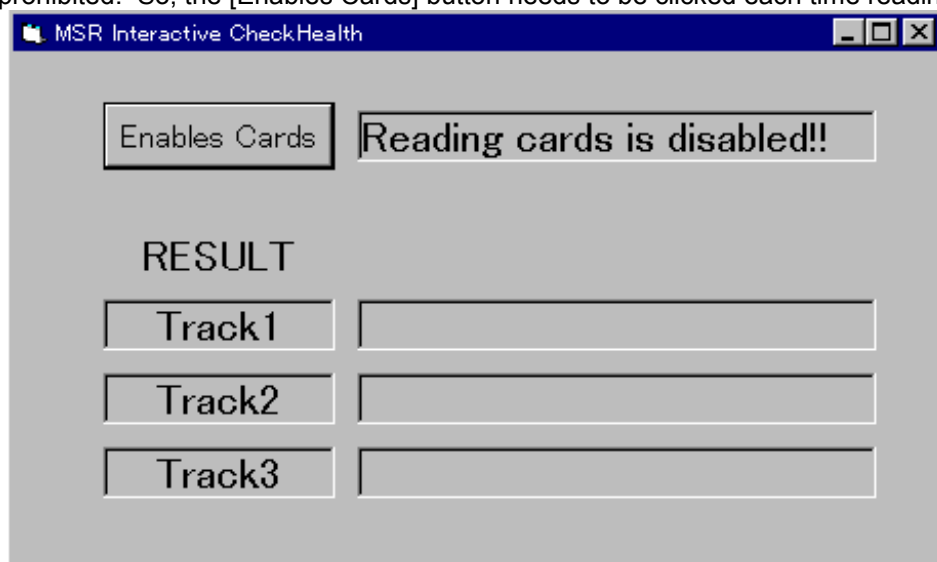
#### 2) External Level

This requests the device to enable reading a card. Reading the card is prohibited if this request succeeds.

| Return Value (ResultCode) | CheckHealthText                | Meaning                                   |
|---------------------------|--------------------------------|---|
| OPOS_SUCCESS              | "External HCheck: Successful"  | Successfully completed.                   |
| OPOS_E_NOHARDWARE         | "External HCheck: Not Support" | Not supported.                            |
| OPOS_E_ILLEGAL            | "External HCheck:Driver Error" | Driver error                              |
| OPOS_E_DISABLED           | "HCheck: Disabled"             | DeviceEnabled=TRUE has not been executed. |
| OPOS_E_NOTCLAIMED         | "Hcheck: Exclusive"            | ClaimDevice has not been executed.        |

#### 3) Interactive Level (OPOS\_CH\_INTERACTIVE)

Displays the following dialog box. After the [Enables Cards] button is clicked and a card is swiped, the data in the card is checked and its result is stored in CheckHealthText. Once a card is read, subsequent reading is prohibited. So, the [Enables Cards] button needs to be clicked each time reading a card.



#### Enable MSR Error Messages

| Message                        | Description  |
|--------------------------------|--|
| Reading card is disabled!!     | Reading card is prohibited as default.<br>Once a card is read, subsequent swiping is prohibited. |
| Read Card!!                    | Reading card is enabled.   |
| Missed enabling reading card!! | Enabling reading card failed.  |

## Track Result Messages

| Message     | Description                             |
|-------------|---|
| Normal End  | Reading data succeeded.                 |
| Start Error | Reading data failed due to Start Error. |
| LCR Error   | Reading data failed due to LRC Error.   |
| No Data     | No data                                 |

| Return Value (ResultCode) | CheckHealthText                    | Meaning   |
|---------------------------|------------------------------------|---|
| OPOS_SUCCESS              | "Interactive<br>Hcheck:Successful" | Successfully completed.                           |
| OPOS_E_FAILURE            | "Interactive Hcheck:Data Error"    | Terminated with an error (Start Error/LRC Error). |
| OPOS_E_ILLEGAL            | "External HCheck:Driver Error"     | Driver error                                      |
| OPOS_E_NOHARDWARE         | "External Hcheck: Not Support"     | Not supported.                                    |
| OPOS_E_DISABLED           | "HCheck: Disabled"                 | DeviceEnabled=TRUE has not been executed.         |
| OPOS_E_NOTCLAIMED         | "Hcheck:Exclusive"                 | ClaimDevice has not been executed.                |

*Note: For ResultCode/CheckHealthText, the last value entered will be retained.*

### 1.1.5 DirectIO Specifications

This Control supports no functions using the DirectIO method.

### 1.1.6 OPOS Registry

The OPOS registry contains the following configuration information:

HKEY\_LOCAL\_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\MSR\MCRSTUB

|             |                                 |
|-------------|---------------------------------|
| General     | "TEC.USBMSR1"                   |
| Service     | "C:\OPOS\TEC\USBMSR.DLL"        |
| Description | "TEC MSR on PKBST-5xU Keyboard" |
| Version     | "1.8"                           |
| DataType    | "MCSRST"   "Standard"           |
| Vkey        | "VK_F13" to "VK_F24" ("VK_F16") |

|             |   |
|-------------|---|
| Service     | Filename of Service Object  |
| Description | Brief explanation of Service Object   |
| Version     | Version number of Service Object  |
| DataType    | Specifies the data type of the card. There are two options as follows:<br>MCSRST: Data format exclusively for the MCSRST.<br>Start sentinel is deleted, and the LRC is converted into 2-byte format.<br>Standard: OPOS APG standard data format<br>Start sentinel, end sentinel, and LRC are deleted.<br>The data type can be selected through the control panel. |
| Vkey        | Specifies the virtual key code which is used for controlling the entry sequence.<br>Selectable from "VK_F13" to "VK_F24"<br>The USB MSR Control uses "VK_F16" as standard, but this is changeable through the control panel.  |

**Table 3 USB PKBST MSR Control – Registries**

### 1.1.7 Limitations and Precautions

#### 1) JIS Type-II Card

When a JIS Type-II card is read, the data will be processed as Track 1 data. ISO Track 1 data will be also stored in the same property. Decoding and division of the field are automatically determined and processed. Although the PKBST-50/51 is equipped with two reading heads, these two heads will not read data at the same time. From the OPOS APG1.5, data of JIS Type-II card is stored in Track4Data property, and also in Track1Data to maintain the compatibility with the lower version of the OPOS APG.

#### 2) Format of TrackxData property

In the case `DataType="MCRST"` is specified in the registry, the format of the TrackxData property will be as follows. In the case of `DataType="Standard"`, the end sentinel and LRC can be deleted.

- |                  |                |
|------------------|----------------|
| (1) Data         | X characters   |
| (2) End sentinel | One character  |
| (3) LRC          | Two characters |

The start sentinel is not stored. The LRC is not one character, but converted 2 characters.

When the LRC is 0x1b on the card, `DecodeData=TRUE` or `Track1Data` is expressed as 0x31,0x62('1','b').

In the case of `DecodeData=FALSE` and non `Track1Data`, the LRC will be the value which cannot be expressed as a character string, but expressed with two characters.

#### 3) Precaution in use of TransmitSentinels property

This property properly operates only when `DataType="Standard"` is specified in the registry. In the case of `DataType="MCRST"`, `CapTransmitSentinels` property is set to `FALSE` because, in the first place, the start sentinel cannot be notified.

Setting `TransmitSentinels` property to `TRUE` enables attaching the start sentinel and end sentinel to the track data stored in `TracknData` property. However, doing this causes the data check not to be performed, and eventually the data including an LRC error or end sentinel error could be stored in `TracknData` property. The application requires to check the data stored in the property for any errors.

Check that the data stored in `TracknData` property starts with the start sentinel and ends with the end sentinel because erroneous data sometimes does not.

#### 4) Notification of Track Level error

The MSR OPOS control may return `OPOS_EL_INPUT_DATA` as `ErrorLocus` of `ErrorEvent` event in case of an input error, only when the notification of the track level error is specified (`ErrorReportingType=MSR_ERT_TRACK`).

In the case the notification of card level error (`ErrorReportingType=MSR_ERT_CARD`) is specified, when any queuing data exists, `ErrorEvent` cannot be received until queued data is input even if an input error occurs. However, `State` property turns to `OPOS_S_ERROR` at the time `ErrorEvent` is put into queue, so it is possible to detect an error state.

#### 5) Auto sensing of the connected port

The USB keyboard is automatically sensed by the Service Object determined according to the open name. However, when two or more units of identical keyboard are connected, which device is sensed is unknown.

#### 6) Plug-and-Play for USB Keyboard in operation

The USB keyboard is a plug-and-play device, but a disconnection and reconnection of the connector during operation is not recommended.

## 7) Notification of the power status

This control determines whether the power status of the keyboard is in "ON LINE" or "OFF or OFFLINE", and notifies of the result. This is substantially corresponding to the disconnection and reconnection of the USB keyboard connector. Though ClaimDevice for unconnected USB device succeeds, subsequent setting of DeviceEnabled property to TRUE results in OPOS\_E\_NOHARDWARE error. Accordingly, the notification of the power status can be started only while the device is connected. The subsequent operations follow the PowerNotify property.

## 8) Operation of TracksToRead property

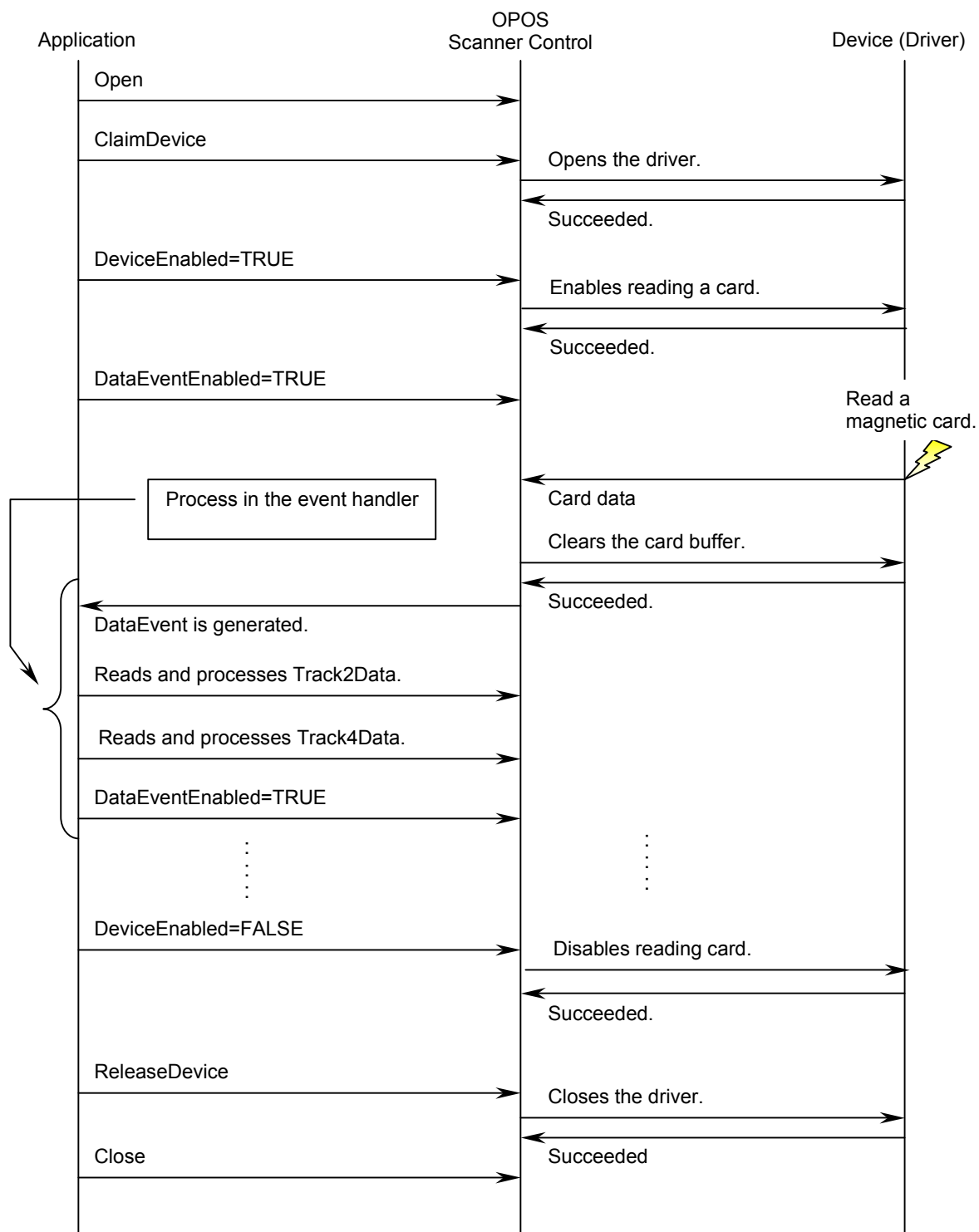
Reading accuracy and reading speed cannot be controlled by this property.

## 9) DeviceEnabled Property

If DeviceEnabled property is operated while a card is read, OPOS\_E\_BUSY is stored in ResultCode property.

### 1.1.8 Usage Example

This section gives a sequence diagram to show an operation flow from reading data on JIS-II track and ISO Track 2 of a magnetic card to a close operation.



### **1.1.9 Log**

A log for this Control is not disclosed.

### 1.1.10 Result When Property/Method is Executed

The OPOS Control notifies the user of a result when a property/method is executed.

The table below shows the ResultCode Property values and OpenResult Property values described in this document.

| ResultCode        | Value |
|-------------------|-------|
| OPOS_SUCCESS      | 0     |
| OPOS_E_CLOSED     | 101   |
| OPOS_E_CLAIMED    | 102   |
| OPOS_E_NOTCLAIMED | 103   |
| OPOS_E_NOSERVICE  | 104   |
| OPOS_E_DISABLED   | 105   |
| OPOS_E_ILLEGAL    | 106   |
| OPOS_E_NOHARDWARE | 107   |
| OPOS_E_OFFLINE    | 108   |
| OPOS_E_NOEXIST    | 109   |
| OPOS_E_EXISTS     | 110   |
| OPOS_E_FAILURE    | 111   |
| OPOS_E_TIMEOUT    | 112   |
| OPOS_E_BUSY       | 113   |
| OPOS_E_EXTENDED   | 114   |

| OpenResult            | Value |
|-----------------------|-------|
| OPOS_OR_ALREADYOPEN   | 301   |
| OPOS_OR_REGBADNAM     | 302   |
| OPOS_OR_REGPROGID     | 303   |
| OPOS_OR_CREATE        | 304   |
| OPOS_OR_BADIF         | 305   |
| OPOS_OR_FAILEDOPEN    | 306   |
| OPOS_OR_BADVERSION    | 307   |
| OPOS_OR_NOPORT        | 401   |
| OPOS_OR_NOTSUPPORTED  | 402   |
| OPOS_OR_CONFIG        | 403   |
| OPOS_OR_SPECIFIC      | 450   |
| OPOS_OR_BADCO         | 451   |
| OPOS_OR_RESOURCEFAIL  | 452   |
| OPOS_OR_ALREADYOPENED | 453   |
|                       |       |

## 1) Results When Property is Executed

The table below gives the results common to all available properties and those unique to certain property.

| Property           | ResultCode        | Meaning  | Error Handling   |
|--------------------|-------------------|--|--|
| Common properties  | OPOS_SUCCESS      | Property setting was completed successfully.   | –  |
|                    | OPOS_E_CLOSED     | The device has been closed.  | Open the device using the Open method, then perform a setting again.   |
| BinaryConversion   | OPOS_E_ILLEGAL    | An invalid value was specified.  | Specify a valid value.   |
| DeviceEnabled      | OPOS_E_NOTCLAIMED | Exclusive access has not been granted.   | Execute the Claim method, then perform a setting again.  |
|                    | OPOS_E_NOHARDWARE | The device is not connected or its power is not turned on.                               | Make sure the device is connected and its power is on, then perform a setting again. If the error occurs again, install the OPOS control and the USB driver again. When the OPOS control and the USB driver have been properly installed, re-install the driver kit suitable for the model of the POS terminal, then perform a setting again. If the error still persists, then investigate the error. |
|                    | OPOS_E_FAILURE    | Enabling reading a card, or setting the beep sound for card reading or read mode failed. | Make sure the device is connected and its power is on, then perform a setting again. If the error occurs again, install the OPOS control and the USB driver again. When the OPOS control and the USB driver have been properly installed, re-install the driver kit suitable for the model of the POS terminal, then perform a setting again. If the error still persists, then investigate the error. |
|                    | OPOS_E_BUSY       | An attempt was made to change the property while a card is being read.                   | Perform a setting again in the idle state.   |
| PowerNotify        | OPOS_E_ILLEGAL    | An invalid value was specified.  | Specify a valid value.   |
|                    |                   | The device is already enabled.   | Unless CapPowerReporting property is OPOS_PR_NONE, the device is already enabled. Set DeviceEnabled property to FALSE, then perform a setting again.   |
|                    |                   | Power notification function is not supported.  | In the case the CapPowerReporting property is OPOS_PR_NONE, the power notification function is not supported.  |
|                    | OPOS_E_NOSERVICE  | The version of the service object is so old that this property is not supported.         | Install the latest service object.   |
| TracksToRead       | OPOS_E_ILLEGAL    | An invalid value was specified.  | Specify a valid value.   |
| ErrorReportingType | OPOS_E_ILLEGAL    | An invalid value was specified.  | Specify a valid value.   |
| TransmitSentinels  | OPOS_E_ILLEGAL    | Notification of the start and end sentinels is not supported.                            | When CapTransmitSentinels property is FALSE, the notification of the start and end sentinels is not supported.   |

| Property | ResultCode        | Meaning  | Error Handling   |
|----------|-------------------|--|--|
|          | OPOS_E_NOHARDWARE | The device is not connected or its power is not turned on.                       | Make sure the device is connected and its power is on, then perform a setting again. If the error occurs again, install the OPOS control and the USB driver again. When the OPOS control and the USB driver have been properly installed, re-install the driver kit suitable for the model of the POS terminal, then perform a setting again. If the error still persists, then investigate the error. |
|          | OPOS_E_FAILURE    | Changing the card read mode of the keyboard failed.                              | Make sure the device is connected and its power is on, then perform a setting again. If the error occurs again, install the OPOS control and the USB driver again. When the OPOS control and the USB driver have been properly installed, re-install the driver kit suitable for the model of the POS terminal, then perform a setting again. If the error still persists, then investigate the error. |
|          | OPOS_E_BUSY       | An attempt was made to change the property while a card is being read.           | Perform a setting again in the idle state.   |
|          | OPOS_E_NOSERVICE  | The version of the service object is so old that this property is not supported. | Install the latest service object.   |

## 2) Results When Open Method is Executed

The Open method differs from other methods and is separately described.

| Return Value     | ResultCode    | OpenResult           | Meaning  | Error Handling  |
|------------------|---------------|----------------------|--|---|
| OPOS_SUCCESS     | OPOS_SUCCESS  | OPOS_SUCCESS         | The device was successfully opened.  | —   |
| OPOS_E_ILLEGAL   | —             | OPOS_OR_ALREADYOPEN  | The device has already been open.  | Make sure the name of the device to be opened is correct.   |
| OPOS_E_NOEXIST   | OPOS_E_CLOSED | OPOS_OR_REG_BADNAME  | The specified device name does not exist in the registry.  | Make sure the name of the device to be opened is correct.   |
|                  |               | OPOS_OR_REGPROGID    | The service object has not been registered correctly.  | Register the driver or service object again.  |
| OPOS_E_NOSERVICE | OPOS_E_CLOSED | OPOS_OR_CREATE       | The service object has not been registered correctly.  | Register the driver or service object again.  |
|                  |               | OPOS_OR_REGPROGID    | The service object has not been registered correctly.  | Register the driver or service object again.  |
|                  |               | OPOS_OR_BADIF        | The service object does not support the methods required.  | Register the service object again.  |
|                  |               | OPOS_OR_FAILEDOPEN   | An error occurred in the service object, but the service object does not support the OpenResult property.    | Register the service object and add the registry again.   |
|                  |               | OPOS_OR_BADVERSION   | The version of the service object is invalid.  | Register the service object again.  |
|                  |               | OPOS_OR_BADCONFIG    | The OPOS registry for the MSR on the USB keyboard does not exist or is not correct.                          | Register the service object and add the registry again  |
|                  |               |                      | Reference to the registry failed.  | If the registry for the MSR on the USB keyboard has been correctly registered, investigate the error. |
|                  |               | OPOS_OR_RESOURCEFAIL | Creation of a resource to serialize the events failed.<br>Creation of a window to process the events failed. | Investigate the error.  |
|                  |               | OPOS_OR_BADCO        | The control object does not support the methods required.  | Register a correct control object again.  |

## 3) Results When A Method Is Executed

The table below describes the result when each method other than the Open method is executed.

| Method                   | Value/ResultCode  | ResultCodeExtended | Meaning  | Error Handling   |
|--------------------------|-------------------|--------------------|--|--|
| Close                    | OPOS_SUCCESS      | —                  | The device was successfully closed.  | —  |
|                          | OPOS_E_CLOSED     | —                  | The device has already been closed.  | —  |
| Claim<br>ClaimDevice     | OPOS_SUCCESS      | —                  | Exclusive access has been granted.   | —  |
|                          | OPOS_E_CLOSED     | —                  | The device has been closed.  | Open the device using the Open method, then execute the method again.                                    |
|                          | OPOS_E_ILLEGAL    | —                  | An invalid timeout parameter value was specified.  | Specify a valid timeout value.   |
|                          |                   | —                  | An event thread was not created.<br>Device input/output thread was not created.  | Investigate the error.   |
|                          | OPOS_E_TIMEOUT    | —                  | While waiting for another application, which has an exclusive access to the device, to release the device, a timeout was called. | Execute the method again after another application releases the exclusive access to the device.          |
| Release<br>ReleaseDevice | OPOS_SUCCESS      | —                  | Exclusive access has been released.  | —  |
|                          | OPOS_E_CLOSED     | —                  | The device has been closed.  | —  |
|                          | OPOS_E_ILLEGAL    | —                  | The application does not have an exclusive access to the applicable device.  | —  |
|                          |                   | —                  | The CheckHealth method is being executed.  | Execute the method again after the CheckHealth method is completed.                                      |
| CheckHealth              | OPOS_SUCCESS      | —                  | A series of health check normally terminated.  | —  |
|                          | OPOS_E_CLOSED     | —                  | The device has been closed.  | Open the device using the Open method, then execute the method again.                                    |
|                          | OPOS_E_NOTCLAIMED | —                  | Exclusive access has not been granted.   | Execute the method again after the Claim method is completed.  |
|                          | OPOS_E_DISABLED   | —                  | The device has been disabled.  | Set the DeviceEnabled property to TRUE, then execute the method again.                                   |
|                          | OPOS_E_ILLEGAL    | —                  | Unsupported health check level was specified. Or, unexpected result was returned from the driver.                                | Check the health check level. If there is no problem with the health check level, investigate the error. |

| Method             | Value/ResultCode  | ResultCodeExtended | Meaning  | Error Handling   |
|--------------------|-------------------|--------------------|--|--|
|                    | OPOS_E_FAILURE    | —                  | An error was found during a series of health check. In the case of the interactive level, this indicates the result of a user's visual check.      | After eliminating the cause of the error, execute the method again. If there is no problem with the device, investigate the error.   |
|                    | OPOS_E_NOHARDWARE | —                  | No response has been returned from the device or the device is not connected.  | Check the connection status. If there are no problems with the connection, re-install the driver kit suitable for the model of the POS terminal, then execute the method again. If the error still persists, investigate the error.  |
| ClearInput         | OPOS_SUCCESS      | —                  | Buffered input of the device was cleared.  | —  |
|                    | OPOS_E_CLOSED     | —                  | The device has been closed.  | Open the device using the Open method, then execute the method again.  |
|                    | OPOS_E_NOTCLAIMED | —                  | Exclusive access has not been granted.   | Execute the Claim method, then execute the method again.   |
| ResetStatistics    | OPOS_E_CLOSED     | —                  | The device has been closed.  | Open the device using the Open method, then execute the method again.  |
|                    | OPOS_E_ILLEGAL    | —                  | Either CapStatisticsReporting property or CapUpdateStatistics property is FALSE. Or, the all specified statistics are undefined or non-resettable. | In the case CapStatisticsReporting property is set to FALSE, the statistics function is not supported.<br>In the case CapUpdateStatistics property is set to FALSE, the statistics information is non-changeable and non-resettable. Otherwise, specify the statistics having the defined name or resettable statistics. |
|                    | OPOS_E_NOSERVICE  | —                  | The version of the service object is so old that the method is not supported.  | Install the latest service object.   |
| RetreiveStatistics | OPOS_E_CLOSED     | —                  | The device has been closed.  | Open the device using the Open method, then execute the method again.  |
|                    | OPOS_E_ILLEGAL    | —                  | The CapStatisticsReporting property is set to FALSE. Or, the all specified statistics are undefined.   | In the case the CapStatisticsReporting property is set to FALSE, the statistics function is not supported. Otherwise, specify the statistics having the defined name.  |
|                    | OPOS_E_NOSERVICE  | —                  | The version of the service object is so old that the method is not supported.  | Install the latest service object.   |
| UpdateStatistics   | OPOS_E_CLOSED     | —                  | The device has been closed.  | Open the device using the Open method, then execute the method again.  |

| Method | Value/ResultCode | ResultCodeExtended | Meaning  | Error Handling   |
|--------|------------------|--------------------|--|--|
|        | OPOS_E_ILLEGAL   | —                  | Either CapStatisticsReporting property or CapUpdateStatistics property is FALSE. Or, the all specified statistics are undefined or non-changeable. | In the case CapStatisticsReporting property is set to FALSE, the statistics function is not supported.<br>In the case CapUpdateStatistics property is set to FALSE, the statistics information is non-changeable and non-resettable. Otherwise, specify the statistics having the defined name or changeable statistics. |
|        | OPOS_E_NOSERVICE | —                  | The version of the service object is so old that the method is not supported.  | Install the latest service object.   |