

No. EAA-02151

[illegible]

APPENDIX

1. Extended Functions of the USB PKBST MSR Control

This section describes the extended functions supported by the USB MSR OPOS Control for MSR on the POS terminal. The extended functions can be used when specified by DirectIO method or the registry.

1.1 DirectIO Specification

The following table shows the extended functions supported by DirectIO method of this control.

Command	Function
TMSR_CMD_SET_CHECKCARD	Specifies the check function for improper data in the card.
TMSR_CMD_GET_CHECKCARD	Obtains the card data check setting.

The constants used for the DirectIO method of this control are defined in *TECMSR.H* provided by TOSHIBA TEC. Please note that this file may be revised when the module is updated. We would recommend you to use the appropriate file for the module to be used.

(1) Specifying the card data check function

Function Generates an LRC error if improper data is contained in card data.

Format	Parameter	Description
	Command	TMSR_CMD_SET_CHECKCARD
	pData	0: Card data check is not performed. 1: Card data check is performed.
	pString	Not used. (Specify a null character string “.”.)

Description Mandatory requirements are Open and Claim.
This function checks whether or not improper data is contained in card data, and if yes, generates an LRC error.
When the card data check is enabled, reading a card containing improper data causes ErrorEvent, which indicates an LRC error, to be generated.
Effective data ranges from 0x20 to 0x7F when decoded into ASCII code. Other data is considered as improper data.

Note This function checks whether character code before a null (0x20) is contained in the card data, and if yes, such data is handled as error data. In the case the program processes error data properly, it is not necessary to use this function.

Return value One of the following values will be returned and also stored in the ResultCode property.

Value	Meaning
OPOS_SUCCESS	The processing succeeded.
OPOS_E_CLOSED	The device has been closed.
OPOS_E_NOTCLAIMED	Exclusive access has not been granted.
OPOS_E_ILLEGAL	The command or pData is wrong. Or pData is NULL.

(2) Obtaining the card data check setting

Function Obtains the setting specified by using the TMSR_CMD_SET_CHECKCARD command.

Format	Parameter	Description
	Command	TMSR_CMD_GET_CHECKCARD
	pData	The current setting is stored in the field specified by this parameter. 0: Card data check is not performed. 1: Card data check is performed.
	pString	Not used. (Specify a null character string “.”.)

Description Mandatory requirement is Open.
This is used to obtain the setting specified by using the
TMSR_CMD_SET_CHECKCARD command.

Note

Return value One of the following values will be returned and also stored in the ResultCode property.

Value	Meaning
OPOS_SUCCESS	The processing succeeded.
OPOS_E_CLOSED	The device has been closed.
OPOS_E_ILLEGAL	The command or pData is wrong. Or pData is NULL.

1.2 OPOS Registry

The OPOS registry contains the following configuration information. The following table shows the extended functions only.

HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\MSR\MCRSTUB

DataType	"MCRST" "Standard" "TransmitSentinels"
StartSentinel	Character string which defines the start sentinel to be added.
Track2AnalysisType	"0" "1"
MSRDataCheck	"0" "1"
DebugLogLevel	"0" "1" "2"
DebugLogFile	"C:\OPOS\TEC\LOG\MCRSTUB.LOG"
ErrorReportingType	"CARD" "TRACK"
ErrorMaskLevel	"0" "1"

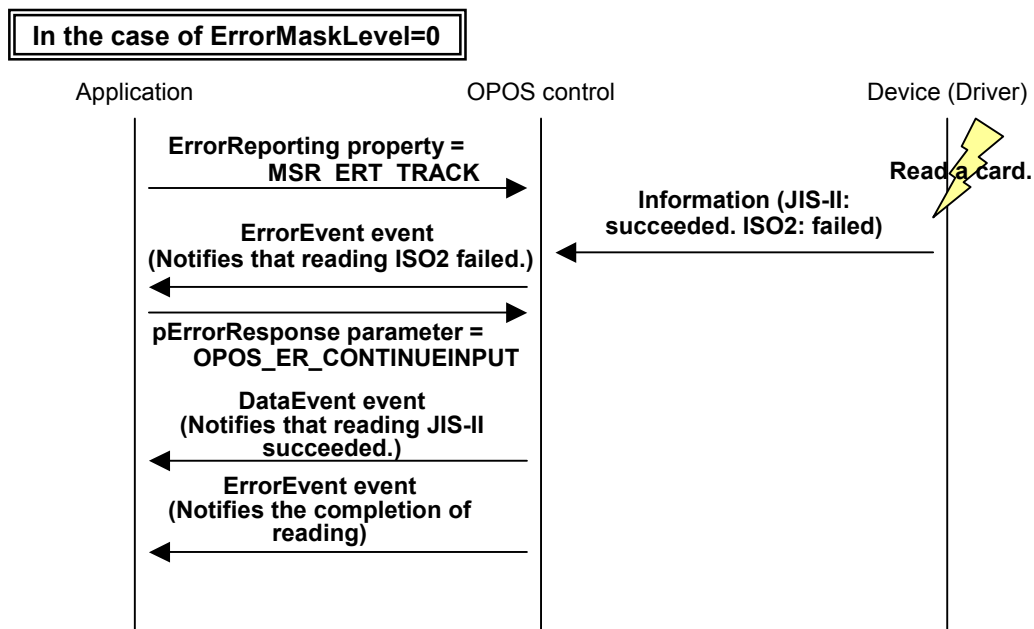
DataType	Specifies the card data formats from the following three types: MCRST: Data format exclusively for the MCRST Start Sentinel is deleted. Standard: OPOS APG standard data format Start/End sentinels and LRC are deleted. TransmitSentinels: Start sentinel is added according to the StartSentinel parameter, and the LRC is deleted. MCRST and Standard can be selected through the control panel.				
StartSentinel	<p>In order to handle a card including special ISO track, it is possible to specify an optional character as the start sentinel in the following format.</p> <p><dlm> track 1 <dlm> track 2 <dlm> track 3 <dlm> Character string used as a separator Track 1 Character to be used as the start sentinel for track 1 Track 2 Character to be used as the start sentinel for track 2 Track 3 Character to be used as the start sentinel for track 3</p> <p>[Example 1] Start sentinel setting: Track 1 = '%', Track 2 = ';', Track 3 = ';' (Same as standard). Separator: '#' "StartSentinel"="#%#;#;"</p> <p>[Example 2] Start sentinel setting: Track 2 = '<' Separator: ',' "StartSentinel"=",,<"</p> <p>[Example 3] Start sentinel setting: Track 1 = '!', Track 3 = '>' Separator: ',' "StartSentinel"=",!,,>"</p> <p>[Note] Characters applicable to the start sentinel are limited as follows: Describe as a value after decoding. Only ISO track is applicable.</p> <table border="1"> <tr> <td>Track 1</td><td>0x20 to 0x5F (ASCII code ' ' to " ")</td></tr> <tr> <td>Track 2 and Track 3</td><td>0x30 to 0x3F (ASCII code '0' to '?')</td></tr> </table>	Track 1	0x20 to 0x5F (ASCII code ' ' to " ")	Track 2 and Track 3	0x30 to 0x3F (ASCII code '0' to '?')
Track 1	0x20 to 0x5F (ASCII code ' ' to " ")				
Track 2 and Track 3	0x30 to 0x3F (ASCII code '0' to '?')				
Track2AnalysisType	Specifies how to process ISO Track 2.				

	<p>0 On the expiration date, obtains the service code only from track 1.</p> <p>1 On the expiration date, obtains the service code from track 1 in priority to other tracks. If no service data exists in track 1, then obtain it from track 2.</p>
MSRDataCheck	<p>Specifies whether to check the card data for any improper data. Proper card data ranges from 0x20 to 0x7F, and if the data is outside of this range, an LRC error results.</p> <p>0 Card data check is not performed.</p> <p>1 Card data check is performed.</p> <p>This parameter can be temporarily changed from the application by using DirectIO method. The registry cannot be changed with this method.</p>
DebugLogLevel	<p>Specifies the extent of the log data to be output. There are three options:</p> <p>“0” The log is not output.</p> <p>“1” Basic log is output.</p> <p>“2” Detailed log is output.</p>
DebugLogFile	<p>Specifies the full pathname of the file where the log is written.</p> <p>If the specified directory is not found, it will not be created automatically.</p>
ErrorReportingType	<p>Specifies the initial value of the error detection level. If this parameter is omitted or an invalid value is set, “CARD” will be effective.</p> <p>The value set here will be the initial value of the ErrorReportingType property.</p> <p>“CARD”: Detects a card level error. Track data is notified only when reading the all track data in the card succeeded. If reading even one track data failed, no data is notified.</p> <p>“TRACK”: Detects a track level error. Track data is notified when reading any one of the track data in the card succeeded. If reading every track data failed, no track data is notified.</p>
ErrorMaskLevel	<p>Specifies the processing when the track of which data reading succeeded and the one failed are mixed in a card. If this parameter is omitted or an invalid value is set, “1” will be effective.</p> <p>This parameter is effective only when the ErrorReportingType property is set to “TRACK”.</p> <p>“0”: Notifies of the information of the track, of which data reading failed, by the ErrorEvent event, and the information of the track, of which data reading succeeded, by DataEvent event.</p> <p>“1”: Notifies only of the information of the track, of which data reading succeeded, by the DataEvent event. The information of the track, of which data reading failed, is not notified.</p>

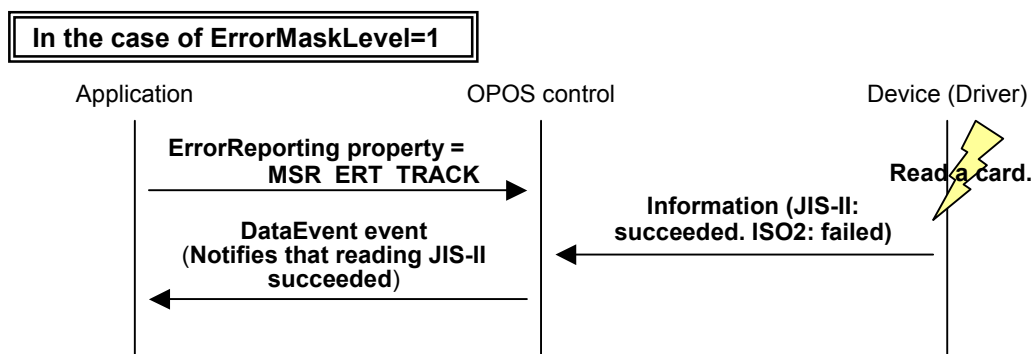
Table 1 Registry of the USB PKBST MCR Control

1.3 Usage Example

This section describes the difference of the operation depending on the ErrorMaskLevel setting in the registry. The following diagram shows an example where a card including JIS-II track and ISO track 2 is read and reading the JIS-II track succeeds and ISO track 2 fails.



In the case of ErrorMaskLevel=0, it is possible to know that reading the ISO2 failed. However, the processing is a bit difficult due to generation of more than one ErrorEvent event.



In the case of ErrorMaskLevel=1, it is impossible to know that reading the ISO2 failed because the notification of the ErrorEvent is not generated. However, the processing is simple since only the information of success in reading the JIS-II is notified by the DataEvent event.

1.4 Result When Property/Method is Executed

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

1) Results When Property is Executed

There is no extended properties.

2) Results When Open Method is Executed

There is no extended Open method functions.

3) Results when DirectIO Method is Executed

Since the result of the DirectIO method differs depending on the commands, they are described separately from the other methods.

Property	Return value/ResultCode	Meaning	Error Handling
Common properties	OPOS_SUCCESS	Processing was completed successfully.	–
	OPOS_E_CLOSED	The device has been closed.	Open the device using the Open method, then perform a setting again.
TMSR_CMD_SET_CHECKCARD	OPOS_E_ILLEGAL	The command is wrong. Or, pData is NULL or the value in the field designated by pData is improper.	Check the command and pData.
	OPOS_E_NOTCLAIMED	Exclusive access has not been granted.	Execute the Claim method, then perform a setting again.
TMSR_CMD_GET_CHECKCARD	OPOS_E_ILLEGAL	The command is wrong. Or, pData is NULL.	Check the command and pData.

4) Results When A Method Is Executed

There is no extended method.

1.5 Log

This Control outputs the trace log and error log according to the registry.
The specification of the log is shown below.

- (1) File name
PKBSTUB.LOG
- (2) Size
Max. 4MB
After the data size reached the maximum, the new data afterwards will overwrite the chronologically oldest one.
- (3) Format
Date, Time, [Function]<Error information>
- (4) Log level
The extent of log data to be output depends on the following log levels.
 - 0: No log is output.
 - 1: Minimum log is output. Error logs are always output when the log level is set to "1".
 - 2: Detailed log is output.
- (5) Log data
No explanation is made in this document.