

Laser Scanner Library Manual

(Version 1.05)

CASIO Computer Co., Ltd.

Copyright ©2011. All rights reserved.

January 2011

Table of the Contents

	Editorial Record	5
Chapter 1.	Overview	7
Chapter 2.	Operation Environment	8
Chapter 3.	Structure	10
3.1	M_TBL	11
Chapter 4.	Functions List	14
4.1	OBROpen	20
4.2	OBRClose	23
4.3	OBRSetLED	24
4.4	OBRGetLED	25
4.5	OBRSetBuzzer	26
4.6	OBRGetBuzzer	27
4.7	OBRSetVibrator	28
4.8	OBRGetVibrator	29
4.9	OBRSetSwingAngle	30
4.10	OBRGetSwingAngle	32
4.11	OBRSetFocus	33
4.12	OBRGetFocus	34
4.13	OBRSetScanningNotification	35
4.14	OBRGetScanningNotification	37
4.15	OBRGetLastEventStatus	38
4.16	OBRSetScanningKey	39
4.17	OBRGetScanningKey	40
4.18	OBRSetAllOptions	41
4.19	OBRGetAllOptions	42
4.20	OBRSetDefaultSymbology	43
4.21	OBRSetDefaultSymbologyEx	44
4.22	OBRSaveLogFile	45
4.23	OBRGetc	46
4.24	OBRGets	48
4.25	OBRGetStatus	50
4.26	OBRClearBuff	51
4.27	OBRSetBuffType	52
4.28	OBRGetBuffType	53
4.29	OBRSetCheckCounter	54
4.30	OBRGetCheckCounter	55
4.31	OBRSetScanningTimeout	56
4.32	OBRGetScanningTimeout	57
4.33	OBRSetSuffixChar	58
4.34	OBRGetSuffixChar	59
4.35	OBRSetScanningType	60
4.36	OBRGetScanningType	61
4.37	OBRSetScanningCounter	62
4.38	OBRGetScanningCounter	63
4.39	OBRSetMultiStepReading	64
4.40	OBRGetMultiStepReading	66
4.41	OBRStartScanning	67

4.42	OBRStopScanning	68
4.43	OBRSetDecodeLevel	69
4.44	OBRGetDecodeLevel	71
4.45	OBRSetScanningCode	72
4.46	OBRGetScanningCode	75
4.47	OBRSetCode39Option	76
4.48	OBRGetCode39Option	78
4.49	OBRSetNW7Option	80
4.50	OBRSetNW7OptionEx	82
4.51	OBRGetNW7Option	84
4.52	OBRSetWPCAddonOption	86
4.53	OBRSetWPCAddonOptionEx	88
4.54	OBRGetWPCAddonOption	90
4.55	OBRSetWPCOption	92
4.56	OBRSetWPCOptionEx	94
4.57	OBRGetWPCOption	96
4.58	OBRSetUPCEAddonOption	98
4.59	OBRSetUPCEAddonOptionEx	100
4.60	OBRGetUPCEAddonOption	102
4.61	OBRSetUPCEOption	104
4.62	OBRSetUPCEOptionEx	106
4.63	OBRGetUPCEOption	108
4.64	OBRSetIDFOption	110
4.65	OBRGetIDFOption	112
4.66	OBRSetITFOption	114
4.67	OBRGetITFOption	116
4.68	OBRSetCode93Option	118
4.69	OBRGetCode93Option	120
4.70	OBRSetCode128Option	122
4.71	OBRGetCode128Option	125
4.72	OBRSetMSIOption	127
4.73	OBRGetMSIOption	129
4.74	OBRSetIATAOption	131
4.75	OBRGetIATAOption	133
4.76	OBRSetRSS14Option	135
4.77	OBRGetRSS14Option	137
4.78	OBRSetRSSLimitedOption	139
4.79	OBRGetRSSLimitedOption	141
4.80	OBRSetRSSExpandedOption	143
4.81	OBRGetRSSExpandedOption	145
4.82	OBRSetRSS14StackedOption	147
4.83	OBRGetRSS14StackedOption	149
4.84	OBRSetRSSExpandedStackedOption	151
4.85	OBRGetRSSExpandedStackedOption	153
4.86	OBRSaveConfigFile	155
4.87	OBRLoadConfigFile	156
4.88	OBRSetFilter	157
4.89	OBRGetFilter	158
4.90	OBRSetFilterOnTimer	159
4.91	OBRGetFilterOnTimer	160

4.92	OBRSetGainControl	161
4.93	OBRGetGainControl	162
4.94	OBRSetBarWidthAdjustment	163
4.95	OBRGetBarWidthAdjustment	164
4.96	OBRSetMarginCheckRatio	165
4.97	OBRGetMarginCheckRatio	167
4.98	OBRSetDecodeLearningMode	168
4.99	OBRGetDecodeLearningMode	169
Chapter 5.	Notes to Programming	170
5.1	Scanning Completion Notification	170
5.2	Configuration File Format	173

No part of this document may be produced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of CASIO Computer Co., Ltd. in Tokyo Japan. Information in this document is subject to change without advance notice. CASIO Computer Co., Ltd. makes no representations or warranties with respect to the contents or use of this manual and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose.

© 2011 CASIO Computer Co., Ltd. All rights reserved.

Editorial Record

Manual Version no.	Date edited	Page	Content
1.00	October 2008		Original
1.01	January 2009	11	In Chapter 3.1, a note is added.
		82	In Chapter 4.52, the description about the parameters is corrected.
		85	In Chapter 4.53, the description about the parameters is corrected.
		88	In Chapter 4.55, the description about the parameters is corrected.
		91	In Chapter 4.56, the description about the parameters is corrected.
		94	In Chapter 4.58, the description about the parameters is corrected.
		97	In Chapter 4.59, the description about the parameters is corrected.
		100	In Chapter 4.61, the description about the parameters is corrected.
		103	In Chapter 4.62, the description about the parameters is corrected.
1.02	March 2009	20	In Chapter 4.1, the description about the parameters is corrected.
		56	In Chapter 4.31, the description about the Device Emulator is corrected.
		60	In Chapter 4.35, the description about the Device Emulator is corrected.
		67	In Chapter 4.41, the description about the return values is corrected.
		82	In Chapter 4.50, the description about the parameters is corrected.
		84	In Chapter 4.51, the description about the parameters is corrected.
		88	In Chapter 4.53, the description about the parameters is corrected.
		94	In Chapter 4.56, the description about the parameters is corrected.
		106	In Chapter 4.62, the description about the parameters is corrected.
		116	In Chapter 4.67, the description about the parameters is corrected.
		120	In Chapter 4.69, the description about the parameters is corrected.

Continue.

1. Overview

The **Laser Scanner Library** provides functions for scanning and decoding scanned bar codes using the integrated laser scanner in the terminal.

The **Laser Scanner Class Library** is a wrapper library layer. The library can be directly manipulated by .NET Compact Framework application.

The use of **Laser Scanner Library** enables the enhancement of source code compatibility for mobile applications regardless of handheld terminal model.

With regard to the function for either the "unsupported error" or "parameter error", at time of process execution, the user is notified that the function is unsupported or that the process itself should be disabled. In this way it is possible to describe a source code that is aware of the "functions" rather than a source code that is aware of the handheld terminal "models" (and the devices integrated in those handheld terminal models), which enables the development of a business application style that attaches "importance to functions".

Note:

The aim of this library is to enhance the source code compatibility for mobile business application, it is not intended to guarantee the compatibility of functions of integrated devices. In actuality it decides whether the "unsupported error" or "parameter error" is required to notify the user that functions are not supported or that the actual process should be disabled.

Also, for the business application to actually run in multiple handheld terminal models, it must be rebuilt to agree with the CPU types (ARMV4T, ARMV4 ...).

2. Operation Environment

Applicable Handheld Terminals

- IT-600 series
- DT-X11 series
- DT-X7 series
- DT-X30 series
- IT-800 series
- IT-300 series
- DT-X8 series

OS

- Microsoft® WindowsCE 5.0
- Microsoft® WindowsCE 6.0
- Microsoft® Windows Mobile 6.1
- Microsoft® Windows Mobile 6.5
- Microsoft® Windows Mobile 6.5.3

Development Environment

- Microsoft® eMbedded C++ Version 4.0 + Service Pack 4
- Microsoft® Visual Studio 2005 + Service Pack 1
- Microsoft® Visual Studio 2008 + Service Pack 1

Supplied Files

- OBReadLib.h
- OBReadLib.lib
- OBReadLib.dll
- OBReadLibNet.dll (Class Library)

Steps to Start Up

For Visual C++:

1. Include the header file, **OBReadLib.h**, in the source program and define it as the library that uses the **OBReadLib.lib** file.
2. **OBReadLib.dll** is already integrated by default in the terminal.

For Visual Basic and Visual C#:

1. Add **OBReadLibNet.dll** to the Project Reference.
2. **OBReadLib.dll** is already integrated by default in the terminal.
3. Copy **OBReadLibNet.dll** into the same folder where the execution module is stored.

Name Space and Class

For **OBReadLibNet.dll** (Class Library), the following classes are available as function and constant references.

Table 2.1

Name Space	Class Name	Description
CaLib	OBReadLibNet.Api	Class for function reference
	OBReadLibNet.Def	Class for constant reference

For detail of the class definition, set up **OBReadLibNet.dll** into reference of Microsoft Visual Studio and check the detail with the object browser.

3. Structure

The following shows the structure provided by the **Laser Scanner Library**.

Table 3.1 Library structure

Structure	Description	Functions used
M_TBL	Structure that stores all the setting information for the scanner driver.	OBRGetAllOptions OBRSetAllOptions

3.1 M_TBL

This structure stores all the setting information for the scanner driver.

```
typedef struct m_tbl{
    DWORD Code;
    BYTE Cd39[6];
    BYTE Nw7[6];
    BYTE Wpcea[6];
    BYTE Wpce[6];
    BYTE Upcea[6];
    BYTE Upce[6];
    BYTE Idsf[6];
    BYTE Itrf[6];
    BYTE Cd93[6];
    BYTE Cd128[6];
    BYTE Msi[6];
    BYTE Iata[6];
    BYTE RSS-14[6];
    BYTE RssLtd[6];
    BYTE RssExp[6];
    BYTE Rss14S[6];
    BYTE RssExpS[6];
    BYTE Resv[14][6];
    BYTE Itrfs[2];
    BYTE Nw7s[2];
    BYTE Cd39s[2];
    BYTE Type;
    BYTE Gain;
    BYTE Buzc;
    BYTE Ledc;
    BYTE BuFc;
    BYTE Endc;
    BYTE Mode;
    BYTE Dumy;
    int cmp_ct;
    int rd_ct;
    int scn_tm;
    DWORD dwFilterMode;
    DWORD dwFilterCnt;
    DWORD dwLearning;
    DWORD dwVibc;
    DWORD dwScanningKey;
    DWORD dwSwing;
    DWORD dwFocus;
    DWORD dwNotification;
    DWORD dwDecodeLevel;
}M_TBL;
```

Table 3.2 M_TBL structure member variables

Member	Description	Functions that use identical setting
Code	Sets up "Enable" or "Disable" for reading bar code symbologies.	OBRSetScanningCode
Cd39[6]	Sets up scanning with Code39 symbology. (Note 1)	OBRSetCode39Option
Nw7[6]	Sets up scanning with NW-7 symbology. (Note 1)	OBRSetNW7Option OBRSetNW7OptionEx
Wpcea[6]	Sets up scanning with WPC Addon read symbology. (Notes 1, 3)	OBRSetWPCAddonOption OBRSetWPCAddonOptionEx
Wpce[6]	Sets up scanning with WPC symbology. (Notes 1, 3)	OBRSetWPCOption OBRSetWPCOptionEx
Upcea[6]	Sets up scanning with UPC-E Addon symbology. (Notes 1, 4)	OBRSetUPCEAddonOption OBRSetUPCEAddonOptionEx
Upce[6]	Sets up scanning with UPC-E symbology. (Notes 1, 4)	OBRSetUPCEOption OBRSetUPCEOptionEx
Idsf[6]	Sets up scanning with IDF symbology. (Note 1)	OBRSetIDFOption
Itrf[6]	Sets up scanning with ITF symbology. (Note 1)	OBRSetITFOption
Cd93[6]	Sets up scanning with Code93 symbology. (Note 1)	OBRSetCode93Option
Cd128[6]	Sets up scanning with Code128 symbology. (Note 1)	OBRSetCode128Option
Msi[6]	Sets up scanning with MSI symbology. (Note 1)	OBRSetMSIOption
Iata[6]	Sets up scanning with IATA symbology. (Note 1)	OBRSetIATAOption
Rss14[6]	Sets up scanning with RSS-14 symbology. (Notes 1, 2)	OBRSetRSS14Option
RssLtd[6]	Sets up scanning with RSS Limited symbology. (Notes 1, 2)	OBRSetRSSLimitedOption
RssExp[6]	Sets up scanning with RSS Expanded symbology. (Notes 1, 2)	OBRSetRSSExpandedOption
Rss14S[6]	Sets up scanning with RSS-14 Stacked symbology. (Notes 1, 2)	OBRSetRSS14StackedOption
RssExpS[6]	Sets up scanning with RSS Expanded Stacked symbology. (Notes 1, 2)	OBRSetRSSExpandedStackedOption
Resv[14][6]	Reserved	-
Itrfs[2]	Reserved	-
Nw7s[2]	Reserved	-
Cd39s[2]	Reserved	-

Continue.

Member	Description	Functions that use identical setting
Type	Sets up read mode.	OBRSetScanningType
Gain	Sets up gain.	OBRSetGainControl
Buzc	Sets up the buzzer.	OBRSetBuzzer
Ledc	Sets up the LED.	OBRSetLED
BuFc	Sets up output mode.	OBRSetBuffType
Endc	Sets up termination code.	OBRSetSuffixChar
Mode	Sets up step read.	OBRSetMultiStepReading
Dumy	Reserved	-
cmp_ct	Sets up the no. of validations.	OBRSetCheckCounter
rd_ct	Sets up the no. of readings.	OBRSetScanningCounter
scn_tm	Sets up timeout.	OBRSetScanningTimeout
dwFilterMode	Sets up noise filter mode.	OBRSetFilter
dwFilterCnt	Sets up a time period between when starting scanning and when starting noise filter operation.	OBRSetFilterOnTimer
dwLearning	Sets up the decoder learning mode.	OBRSetDecodeLearningMode
dwVibe	Sets up the vibrator.	OBRSetVibrator
dwScanningKey	Sets up scanning trigger key.	OBRSetScanningKey
dwSwing	Sets up laser beam swing angle mode.	OBRSetSwingAngle
dwFocus	Sets up laser focus.	OBRSetFocus
dwNotification	Sets up notification of scanning complete event.	OBRSetScanningNotification
dwDecodeLevel	Sets up decoding level.	OBRSetDecodeLevel

Notes:

- The parameters are as follows.
 - [0] Reserved
 - [1] Minimum no. of digits.
 - [2] Maximum no. of digits.
 - [3] Specify output format.
 - [4] Specify check digit calculation.
 - [5] Specify check character output.
- The RSS symbologies change its name to "GS1 DataBar".
 - RSS-14 → GS1 DataBar Omnidirectional
 - RSS Limited → GS1 DataBar Limited
 - RSS Expanded → GS1 DataBar Expanded
 - RSS-14 Stacked → GS1 DataBar Stacked
 - RSS Expanded Stacked → GS1 DataBar Expanded Stacked
- To set the Check Digit Calculation to "Disable" for WPC Addon symbology or WPC symbology, be sure to set the Check Digit Calculation to "Disable" for both WPC Addon and WPC symbologies.
- To set the Check Digit Calculation to "Disable" for UPC-E Addon symbology or UPC-E symbology, be sure to set the Check Digit Calculation to "Disable" for both UPC-E Addon and UPC-E symbologies.

4. Functions List

Table 4.1 Basic functions

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
OBROpen	Sets up scanning enable status.	Y	Y	Y	Y	-	Y	Y	Y
OBRClose	Sets up scanning disable status.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetLED	Sets up LED notification.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetLED	Retrieves status of LED notification.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetBuzzer	Sets up buzzer notification.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetBuzzer	Retrieves status of buzzer notification.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetVibrator	Sets up vibrator notification.	-	Y	Y	-	-	Y	Y	Y
OBRGetVibrator	Retrieves status of vibrator notification.	-	Y	Y	-	-	Y	Y	Y
OBRSetSwingAngle	Sets up swing angle control mode.	-	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetSwingAngle	Retrieves status of swing angle control mode.	-	Y S1	Y S1	Y	-	Y	Y	Y
OBRSetFocus	Sets up laser focus.	-	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetFocus	Retrieves status of laser focus.	-	Y S1	Y S1	Y	-	Y	Y	Y
OBRSetScanningNotification	Sets up scanning complete notification.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetScanningNotification	Retrieves scanning complete notification.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetLastEventStatus	Retrieves last event status.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetScanningKey	Sets up Trigger keys.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetScanningKey	Retrieves status of Trigger keys.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetAllOptions	Sets up laser scanner driver.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetAllOptions	Retrieves status of laser scanner driver.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetDefaultSymbology	Sets up default status of laser scanner driver.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetDefaultSymbologyEx	Resumes all the settings to the factory-defaults.	Y	Y	Y	Y	-	Y	Y	Y
OBRSaveLogFile	Outputs driver log information.	Y	Y	Y	Y	-	Y	Y	Y

Table 4.2 Decoding functions

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
OBRGetc	Retrieves one character from OBR buffer.	Y	Y	Y	Y	-	Y	Y	Y
OBRGets	Reads character string for one bar code from decoding data storage buffer.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetStatus	Retrieves OBR buffer status.	Y	Y	Y	Y	-	Y	Y	Y
OBRClearBuff	Clears OBR buffer.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetBuffType	Sets up decoding data output mode.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetBuffType	Retrieves decoding data output mode.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetCheckCounter	Sets up the no. of validation for decoding data.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetCheckCounter	Retrieves the no. of validations for decoding data.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetScanningTimeout	Sets up a time period of timeout for scanning complete.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetScanningTimeout	Retrieves timeout of scanning complete.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetSuffixChar	Sets up suffix control code appended to decoding data.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetSuffixChar	Retrieves suffix control code appended to decoding data.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetScanningType	Sets up scanning mode.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetScanningType	Retrieves scanning mode.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetScanningCounter	Sets up the no. of times for continuous scanning.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetScanningCounter	Retrieves the no. of times for continuous scanning.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetMultiStepReading	Sets up the no. of bar codes to scan in multi-step scanning mode.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetMultiStepReading	Retrieves the no. of bar codes to scan in multi-step scanning mode.	Y	Y	Y	Y	-	Y	Y	Y
OBRStartScanning	Emits laser beam and start scanning a bar code.	Y	Y	Y	Y	-	Y	Y	Y
OBRStopScanning	Stops emitting laser beam and scanning a bar code.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetDecodeLevel	Sets up decoding level.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetDecodeLevel	Retrieves decoding level.	Y	Y	Y	Y	-	Y	Y	Y

Table 4.3 Symbology functions

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
OBRSetScanningCode	Sets up scanning with each bar code symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetScanningCode	Retrieves scanning status with each bar code symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetCode39Option	Sets up scanning with Code39 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetCode39Option	Retrieves decode options of Code39 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetNW7Option	Sets up scanning with NW-7 code symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetNW7OptionEx	Sets up scanning with NW-7 code symbology.	-	Y S2	Y	Y	-	Y	Y	Y
OBRGetNW7Option	Retrieves decode options of NW-7 code symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetWPCAddonOption	Sets up scanning with WPC Addon symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetWPCAddonOptionEx	Sets up scanning with WPC Addon symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetWPCAddonOption	Retrieves decode options of WPC Addon symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetWPCOption	Sets up scanning with WPC symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetWPCOptionEx	Sets up scanning with WPC symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetWPCOption	Retrieves decode options of WPC symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetUPCEAddonOption	Sets up scanning with UPC-E Addon symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetUPCEAddonOptionEx	Sets up scanning with UPC-E Addon symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetUPCEAddonOption	Retrieves decode options of UPC-E Addon symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetUPCEOption	Sets up scanning with UPC-E symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetUPCEOptionEx	Sets up scanning with UPC-E symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetUPCEOption	Retrieves decode options of UPC-E symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetIDFOption	Sets up scanning with Industrial 2of5 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetIDFOption	Retrieves decode options of Industrial 2of5 symbology.	Y	Y	Y	Y	-	Y	Y	Y

Continue.

OBRSetITFOption	Sets up scanning with ITF (Interleaved 2of5) symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetITFOption	Retrieves decode options of ITF (Interleaved 2of5) symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetCode93Option	Sets up scanning with Code93 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetCode93Option	Retrieves decode options of Code93 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetCode128Option	Sets up scanning with Code128 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetCode128Option	Retrieves decode options of Code128 symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetMSIOption	Sets up scanning with MSI symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetMSIOption	Retrieves decode options of MSI symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetIATAOption	Sets up scanning with IATA symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetIATAOption	Retrieves decode options of IATA symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetRSS14Option	Sets up scanning with RSS-14 symbology.	-	Y	Y	Y	-	Y	Y	Y
OBRGetRSS14Option	Retrieves decode options of RSS-14 Stacked symbology.	-	Y	Y	Y	-	Y	Y	Y
OBRSetRSSLimitedOption	Sets up scanning with RSS Limited symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetRSSLimitedOption	Retrieves decode options of RSS Limited symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetRSSExpandedOption	Sets up scanning with RSS Expanded symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRGetRSSExpandedOption	Retrieves decode options of RSS Expanded symbology.	Y	Y	Y	Y	-	Y	Y	Y
OBRSetRSS14StackedOption	Sets up scanning with RSS-14 Stacked symbology.	-	C	Y	Y	-	Y	Y	Y
OBRGetRSS14StackedOption	Retrieves decode options of RSS-14 Stacked symbology.	-	C	Y	Y	-	Y	Y	Y
OBRSetRSSExpandedStackedOption	Sets up scanning with RSS Expanded Stacked symbology.	-	C	Y	Y	-	Y	Y	Y
OBRGetRSSExpandedStackedOption	Retrieves decode options of RSS Expanded Stacked symbology.	-	C	Y	Y	-	Y	Y	Y

Table 4.4 Save and Load functions

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
OBRSaveConfigFile	Saves all settings for the scanner driver into configuration file.	Y	Y	Y	Y	-	Y	Y	Y
OBRLoadConfigFile	Reads and then loads scanner configuration file into the scanner driver.	Y	Y	Y	Y	-	Y	Y	Y

Table 4.5 Filter functions

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
OBRSetFilter	Sets up noise filter mode.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetFilter	Retrieves noise filter mode.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRSetFilterOnTimer	Sets up start time for noise filter.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetFilterOnTimer	Retrieves start time for noise filter.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRSetGainControl	Sets up gain control.	-	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetGainControl	Retrieves gain control.	-	Y S1	Y S1	Y	-	Y	Y	Y

Table 4.6 Learning functions

Function	Description	DT-X11	IT-600	DT-X7	DT-X30	IT-3100	IT-800	IT-300	DT-X8
OBRSetBarWidthAdjustment	Sets up the adjustment of thickness of bars.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetBarWidthAdjustment	Retrieves the adjustment set for the thickness of bars.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRSetMarginCheckRatio	Sets up a threshold value for right and left side space margin.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetMarginCheckRatio	Retrieves the threshold value for right and left side space margin.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRSetDecodeLearningMode	Sets up the decoder learning mode.	Y	Y S1	Y S1	Y	-	Y	Y	Y
OBRGetDecodeLearningMode	Retrieves the decoder learning mode.	Y	Y S1	Y S1	Y	-	Y	Y	Y

Denotations:

Y : The models supports the function.

- : Unsupported error responses when the function is called.

C : IT-600 with either its OS ver. 5044 and Service Pack 1.02, or its OS ver. 5069, or a later version preinstalled supports the function.

S1: IT-600M30U, IT-600M30UC and DT-X7M10U do not support the function. Other models of the respective series support the function.

S2: IT-600M30 does not support the function. Other models of the series support the function.

4.1 OBROpen

This function sets up the scanner driver effect. The enabled state of the scanner driver will be kept effect until when **OBRClose** function is carried out. While the Laser Scanner Library is being opened, scanning will start when Trigger key is pressed. During this period, various settings for scanning cannot be changed. To make any change on the settings, first call **OBRClose** function to close the Laser Scanner Library, and then change settings. Because the Trigger key is occupied by the library during this period, the user cannot access it.

Calling Sequences

```
[C++]
int OBROpen(
    HWND hWnd,
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBROpen( _
    ByVal hWnd As IntPtr, _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBROpen(
    IntPtr hWnd,
    Int32 dwMode
);
```

Parameters

hWnd

This parameter is for specifying an application window handle. In case where the decoded data output mode is set to "Key Buffer Output", the data will be output to a specified window handle. With application that does not have a window handle, set the decoded data output mode to other mode not "Key Buffer Output", and specify "NULL" for this parameter. The output mode can be set with **OBRSetBuffType** function.

dwMode

This parameter is for setting "Enable" or "Disable" for each bar code symbology which is specified with a sum of the values in logical OR operation selected in the list below. These are designated using the following OR (logical add) values. If zero "0" is specified, scanning condition is set enabled with the current settings. Or, if any value other than 0 is set, first all the settings will be changed to the default settings, and then settings with the specified values will become reflected. See note 5.

OBR_CD39	: Code39 symbology enabled
OBR_NW_7	: NW-7 symbology enabled
OBR_WPCA	: WPC Addon symbology enabled
OBR_WPC	: WPC symbology enabled
OBR_UPEA	: UPC-E Addon symbology enabled
OBR_UPE	: UPC-E symbology enabled
OBR_IDF	: Industrial 2of5 symbology enabled
OBR_ITF	: Interleaved 2of5 symbology enabled
OBR_CD93	: Code93 symbology enabled
OBR_CD128	: Code128 symbology enabled (note 7)
OBR_MSI	: MSI symbology enabled
OBR_IATA	: IATA symbology enabled
OBR_RSS14	: RSS-14 symbology enabled (notes 1 and 6)
OBR_RSSLTD	: RSS Limited symbology enabled (notes 1 and 6)
OBR_RSSEXP	: RSS Expanded symbology enabled (notes 1 and 6)
OBR_RSS14S	: RSS-14 stacked symbology enabled (notes 4 and 6)
OBR_RSSEXP_S	: RSS Expanded stacked symbology enabled (notes 4 and 6)
OBR_ALL	: All the symbologies enabled
OBR_CHK_ON	: Check digit calculation enabled for all the symbologies (note 2)
OBR_OUT_ON	: Check character output enabled for all the symbologies (note 3)

Notes:

1. Available only with symbologies that support check digit calculation.
2. Available only with symbologies that support check character output.
3. IT-600 and DT-X11 support the check character output.
4. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that the settings for reading Industrial 2of5 and IATA symbologies are to be concurrently made enabled. If it is done wrongly, all other settings for reading other symbologies are disabled and a parameter error is returned.
5. Individual settings for reading RSS-14, RSS-14 Stacked, RSS Expanded, and RSS Expanded Stacked symbologies are not supported. If setting for reading a standard RSS symbology is set enabled, all the corresponded RSS Stacked symbologies are automatically set enabled for reading the symbologies.
6. The RSS symbologies change its name to "GS1 DataBar".
 - RSS-14 → GS1 DataBar Omnidirectional
 - RSS Limited → GS1 DataBar Limited
 - RSS Expanded → GS1 DataBar Expanded
 - RSS-14 Stacked → GS1 DataBar Stacked
 - RSS Expanded Stacked → GS1 DataBar Expanded Stacked

7. The setting for reading Code128 symbology effect will also make reading the GS1-128 symbology (previously named as EAN-128 symbology) possible.

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error
OBR_ERROR_HOTKEY	: Scan key registration error. In the Device Emulator, this value is not returned.
OBR_ERROR_INVALID_ACCESS	: Concurrent use error with digital camera,. In the Device Emulator, this value is not returned.
OBR_PRM	: Parameter error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.2 OBRClose

This function disables the scanning operations and the laser scanner driver.

Calling Sequences

```
[C++]  
int OBRClose( )
```

```
[Visual Basic]  
Public Shared Function OBRClose() As Int32
```

```
[C#]  
public static Int32 OBRClose()
```

Parameter

None

Return Values

OBR_OK	: Normal end
FUNCTION_UN SUPPORT	: Unsupported error

4.3 OBRSetLED

This function sets up notification mode for the completion of scanning bar code using the integrated LED (Indicator #1 or #2, model dependant).

Calling Sequences

```
[C++]
int OBRSetLED(
    BYTE byMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetLED( _
    ByVal byMode As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetLED(
    Byte byMode
);
```

Parameters

byMode

This parameter is for specifying an LED notification mode selecting one of the values listed below.

OBR_LEDOFF	: No LED control
OBR_LEDON	: LED control
OBR_LEDEROF	: LED control (no control if error in scanning.)

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.4 OBRGetLED

This function retrieves the status of notification mode with the integrated LED.

Calling Sequences

```
[C++]
int OBRGetLED(
    BYTE *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetLED( _
    ByRef pMode As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetLED(
    ref Byte pMode
);
```

Parameters

pMode

This is for retrieving the value of the LED notification mode. See **OBRSetLED** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.5 OBRSetBuzzer

This function sets up the notification mode for scanning complete with the integrated buzzer.

Calling Sequences

```
[C++]
int OBRSetBuzzer(
    BYTE byMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetBuzzer( _
    ByVal byMode As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetBuzzer(
    Byte byMode
);
```

Parameters

byMode

This parameter is for specifying buzzer notification mode selecting either of the values listed below.

OBR_BUZOFF	: Disable the buzzer notification
OBR_BUZON	: Enable the buzzer notification

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.6 OBRGetBuzzer

This function retrieves the buzzer notification mode.

Calling Sequences

```
[C++]
int OBRGetBuzzer(
    BYTE *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetBuzzer( _
    ByRef pMode As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetBuzzer(
    ref Byte pMode
);
```

Parameters

pMode

This parameter is for retrieving the value of the buzzer notification mode. See **OBRSetBuzzer** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.7 OBRSetVibrator

This function sets up the notification mode for scanning complete with the integrated vibrator.

Calling Sequences

```
[C++]
int OBRSetVibrator(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetVibrator( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetVibrator(
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying vibrator notification mode selecting either of the values listed below.

OBR_VIBOFF	: Disable the vibration notification mode
OBR_VIBON	: Enable the vibration notification mode

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.8 OBRGetVibrator

This function retrieves the notification mode for scanning complete with the integrated vibrator.

Calling Sequences

```
[C++]
int OBRGetVibrator(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetVibrator( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetVibrator(
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is for specifying the value for the vibration notification mode. See **OBRSetVibrator** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.9 OBRSetSwingAngle

This function sets up the Scan Width Control. The laser beam irradiation width for each angle can be set in the OBR calibration. If the setting is not performed, the irradiation width will be automatically set with the factory default setting.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetSwingAngle** function.

Calling Sequences

```
[C++]
int OBRSetSwingAngle(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetSwingAngle( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetSwingAngle(
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying laser beam swing angle selecting one of the values listed below.

OBR_SWING_OFF	: No swing angle control
OBR_SWING_WIDE	: Wide swing angle
OBR_SWING_MIDDLE	: Middle swing angle
OBR_SWING_NARROW	: Narrow swing angle

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_PRM	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

The OBR calibration can be performed using the dedicated bar image. Navigate to **Control Panel** → **Laser Setting** icon and then **Others** tab.

Setting up the scan width control function (OBRSetSwingAngle function) in one of four modes while the setting gain (OBRSetGainControl function) is set up in Auto mode may

cause a deterioration of scanning bar codes that are printed in a poor quality. If you need to set up the scan width control function, select the setting gain in Mode 0 to Mode 3. Or, if you select the setting gain in Auto mode, select the scan width control function in "No control on laser beam emission width" mode only.

4.10 OBRGetSwingAngle

This function retrieves the Scan Width Control mode.

Calling Sequences

```
[C++]
int OBRGetSwingAngle(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetSwingAngle( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetSwingAngle(
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is for retrieving the Scan Width Control mode. See **OBRGetSwingAngle** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.11 OBRSetFocus

This function sets up the Scan Verification Beam mode. The Scan Verification Beam is a function that focuses a scanned bar code after its scanning is succeeded so that the bar code can be distinguished from others in the same vicinity to avoid confusion by the user.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetFocus** function.

Calling Sequences

```
[C++]
int OBRSetFocus(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetFocus( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetFocus(
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying Scan Verification Beam mode selecting either of the values listed below.

OBR_FOCUSOFF	: Disable the Scan Verification Beam mode.
OBR_FOCUSON	: Enable the Scan Verification Beam mode.

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_PRN	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.12 OBRGetFocus

This function retrieves the current Scan Verification Beam mode.

Calling Sequences

```
[C++]  
int OBRGetFocus(  
    DWORD *pMode  
)
```

```
[Visual Basic]  
Public Shared Function OBRGetFocus( _  
    ByRef pMode As Int32 _  
) As Int32
```

```
[C#]  
public static Int32 OBRGetFocus(  
    ref Int32 pMode  
) ;
```

Parameters

pMode

This parameter is for retrieving the Scan Verification Beam mode. See **OBRSetFocus** for values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.13 OBRSetScanningNotification

This function sets up the notification method for scanning complete via either "Event" or "Window message".

- Notification via Window Message
Window message of WM_OBR_SCANNING(WM_USER+0x530) is issued to the specified window handle.
- Notification via Event
Event generated when scanning is complete is "OBRScanningEvent". Since the name is expressed in Unicode in WindowsCE, it must be specified as TEXT("OBRScanningEvent") in application program.

See Chapter 5.1 "Scanning Completion Notification for detail.

Calling Sequences

```
[C++]
int OBRSetScanningNotification(
    DWORD dwMode,
    HWND hWnd
)
```

```
[Visual Basic]
Public Shared Function OBRSetScanningNotification( _
    ByVal dwMode As Int32, _
    ByVal hWnd As IntPtr _
) As Int32
```

```
[C#]
public static Int32 OBRSetScanningNotification(
    Int32 dwMode,
    IntPtr hWnd
);
```

Parameters

dwMode

This parameter is for specifying notification mode selecting either of the values listed below.

OBR_MESSAGE	: Notification via window message
OBR_EVENT	: Notification via Event

hWnd

This parameter is for specifying handle to which the window message is sent in case where OBR_MESSAGE is specified in *dwMode* parameter.

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_PRM	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.14 OBRGetScanningNotification

This function retrieves the scanning completion notification. The setting status for scanning complete notification via either "Event" or "Window message" is retrieved.

Calling Sequences

```
[C++]
int OBRGetScanningNotification(
    DWORD *pMode,
    HWND *hWnd
)
```

```
[Visual Basic]
Public Shared Function OBRGetScanningNotification( _
    ByRef pMode As Int32, _
    ByRef hWnd As IntPtr _
) As Int32
```

```
[C#]
public static Int32 OBRGetScanningNotification(
    ref Int32 pMode,
    ref IntPtr hWnd
);
```

Parameters

pMode

This parameter is for retrieving the notification mode. See **OBRSetScanningNotification** function for the values to retrieve.

hWnd

This parameter is for retrieving the handle of window message when OBR_MESSAGE has been set in **pMode* parameter.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.15 OBRGetLastEventStatus

If the scanning completion notification has been set to "via Event", this function retrieves the factor that forced scanning a bar code to end. See Chapter 5.1 "Scanning Completion Notification" for detail.

Calling Sequences

```
[C++]
int OBRGetLastEventStatus(
    DWORD *pStatus
)
```

```
[Visual Basic]
Public Shared Function OBRGetLastEventStatus( _
    ByRef pStatus As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetLastEventStatus(
    ref Int32 pStatus
);
```

Parameters

pStatus

This parameter is for retrieving a factor that forces the scanning to end.

OBR_NONE	: No event factor
OBR_SUCCESS	: Scanning succeeded
OBR_TIMEOUT	: Timeout
OBR_BUFFEROVER	: OBR buffer in full error

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.16 OBRSetScanningKey

This function sets up scanning Trigger keys. While the scanner driver is being set opened with **OBROpen** function, scanning starts when one of the Trigger keys specified with **OBRSetScanningKey** function is pressed.

Calling Sequences

```
[C++]
int OBRSetScanningKey(
    DWORD dwKey
)
```

```
[Visual Basic]
Public Shared Function OBRSetScanningKey( _
    ByVal dwKey As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetScanningKey(
    Int32 dwKey
);
```

Parameters

dwKey

This parameter is for specifying a sum of the values in logical OR for the Trigger keys selected in the list below.

OBR_TRIGGERKEY_L	: L Trigger key
OBR_TRIGGERKEY_R	: R Trigger key
OBR_CURSORKEY_L	: Cursor left key
OBR_CURSORKEY_R	: Cursor right key
OBR_CURSORKEY_U	: Cursor-up key
OBR_CURSORKEY_D	: Cursor-down key
OBR_GUNTRIGGER	: Grip trigger key
OBR_GUNGRIP	: Grip trigger key
OBR_CENTERTRIGGER	: Center key

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_PRM	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.17 OBRGetScanningKey

This function retrieves the current scanning Trigger keys. While the scanner driver is being set opened with **OBROpen** function, scanning starts when one of the Trigger keys specified with **OBRSetScanningKey** function is pressed. During this period, all the scanning Trigger keys set are occupied by the scanner driver, the user cannot access unless **OBRClose** function is carried out to make the keys free from the scanner driver.

Calling Sequences

```
[C++]
int OBRGetScanningKey(
    DWORD *pKey
)
```

```
[Visual Basic]
Public Shared Function OBRGetScanningKey( _
    ByRef pKey As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetScanningKey(
    ref Int32 pKey
);
```

Parameters

pKey

This parameter is for retrieving the scanning Trigger keys. See **OBRSetScanningKey** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.18 OBRSetAllOptions

This function sets up all the settings for the scanner driver at a time.

Calling Sequences

```
[C++]
int OBRSetAllOptions(
    M_TBL *pModeTable
)
```

```
[Visual Basic]
Public Shared Function OBRSetAllOptions( _
    ByVal pModeTable As M_TBL _
) As Int32
```

```
[C#]
public static Int32 OBRSetAllOptions(
    M_TBL pModeTable
);
```

Parameters

pModeTable

This parameter is for specifying the scanner driver setting structure.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

Note:

See the M_TBL structure for detail.

4.19 OBRGetAllOptions

This function retrieves all the settings at a time for the scanner driver.

Calling Sequences

```
[C++]
int OBRGetAllOptions(
    M_TBL *pModeTable
)
```

```
[Visual Basic]
Public Shared Function OBRGetAllOptions( _
    ByVal pModeTable As M_TBL _
) As Int32
```

```
[C#]
public static Int32 OBRGetAllOptions(
    M_TBL pModeTable
);
```

Parameters

pModeTable

This parameter is for specifying pointer to the structure that saves the retrieved scanner driver settings.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

Note:

See the M_TBL structure for detail.

4.20 OBRSetDefaultSymbology

This function sets up all the settings of the scanner driver to the defaults. Under the condition that the configuration file exists and **OBRLoadConfigFile** function has been called, the settings in the configuration file will be set to the respective defaults with this function.

Calling Sequences

```
[C++]  
int OBRSetDefaultSymbology( )
```

```
[Visual Basic]  
Public Shared Function OBRSetDefaultSymbology() As Int32
```

```
[C#]  
public static Int32 OBRSetDefaultSymbology()
```

Parameter

None

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.21 OBRSetDefaultSymbologyEx

This function resumes all the settings of the scanner driver to the factory-defaults.

In the Device Emulator, the function returns always FUNCTION_UN SUPPORT.

Calling Sequences

```
[C++]  
int OBRSetDefaultSymbologyEx( )
```

```
[Visual Basic]  
Public Shared Function OBRSetDefaultSymbologyEx() As Int32
```

```
[C#]  
public static Int32 OBRSetDefaultSymbologyEx()
```

Parameter

None

Return Values

OBR_OK	: Normal end
FUNCTION_UN SUPPORT	: Unsupported error

4.22 OBRSaveLogFile

This function outputs log information about the scanner driver saved in the memory by the scanner driver to the specified log file.

Calling Sequences

```
[C++]  
int OBRSaveLogFile( )
```

```
[Visual Basic]  
Public Shared Function OBRSaveLogFile() As Int32
```

```
[C#]  
public static Int32 OBRSaveLogFile()
```

Parameter

None

Return Values

OBR_OK	: Normal end
FUNCTION_UNSupport	: Unsupported error

Notes:

- With the following file names log information is output to the driver log and route directory.
Information about the scanner control : OBRLog.dat
Information about the decoder : DecodeLog.dat
- Before calling **OBRSaveLogFile** function, be sure to open the scanner driver with **OBROpen** function.
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that the decoder related log is output.

4.23 OBRGetc

This function retrieves data for one character from the decoding data storage buffer in the scanner driver.

In the Device Emulator, the function retrieves the code specified by the I/O Simulator.

Calling Sequences

```
[C++]
int OBRGetc(
    DWORD *dwRcd
)
```

```
[Visual Basic]
Public Shared Function OBRGetc( _
    ByRef dwRcd As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetc(
    ref Int32 dwRcd
);
```

Parameters

dwRcd

This parameter is for retrieving a symbology of retrieved decoding data.

OBR_NONDT	: No data
OBR_CD39	: Code39 symbology
OBR_NW_7	: NW-7 symbology
OBR_WPCA	: WPC (except UPC-E) Addon symbology
OBR_WPC	: WPC (except UPC-E) symbology
OBR_UPEA	: UPC-E Addon symbology
OBR_UPE	: UPC-E symbology
OBR_IDF	: Industrial 2of5 symbology
OBR_ITF	: Interleaved 2of5 symbology
OBR_CD93	: Code93 symbology
OBR_CD128	: Code128 symbology (note 4)
OBR_MSI	: MSI symbology
OBR_IATA	: IATA symbology
OBR_RSS14	: RSS-14 symbology or RSS-14 Stacked symbology. (notes 2 and 3)
OBR_RSSLTD	: RSS Limited symbology. (notes 2 and 3)
OBR_RSSEXP	: RSS Expanded symbology or RSS Expanded Stacked symbology. (notes 2 and 3)

Return Values

OBR data (for one character)

otherwise

OBR_NOT_DEVICE : OBR (the scanner unit) in abnormal state. In the Device Emulator, this value is not returned.

FUNCTION_UNSUPPORTED : Unsupported error

Notes:

1. Carrying out **OBRGetc** function without opening the scanner driver first may result in abnormal state.
2. IT-600, DT-X7, DT-X11, DT-X30, IT-800, IT-300, and DT-X8 support the symbology.
3. The RSS symbologies change its name to "GS1 DataBar".
 - RSS-14 → GS1 DataBar Omnidirectional
 - RSS Limited → GS1 DataBar Limited
 - RSS Expanded → GS1 DataBar Expanded
 - RSS-14 Stacked → GS1 DataBar Stacked
 - RSS Expanded Stacked → GS1 DataBar Expanded Stacked
4. The function retrieves "OBR_CD128" even if reading the GS1-128 symbology (previously named as EAN128 symbology).

4.24 OBRGets

This function retrieves decoded data for one character from the decoding data storage buffer in the scanner driver.

In the Device Emulator, the function retrieves the code specifies by the I/O Simulator.

Calling Sequences

```
[C++]
int OBRGets(
    BYTE *pDecodeData,
    DWORD *pCode,
    BYTE *pLength
)
```

```
[Visual Basic]
Public Shared Function OBRGets( _
    ByVal pDecodeData As Byte(), _
    ByRef pCode As Int32, _
    ByRef pLength As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGets(
    Byte[] pDecodeData,
    ref Int32 pCode,
    ref Byte pLength
);
```

Parameters

pDecodeData

This parameter is for retrieving the decoded data.

pCode

This parameter is for retrieving the symbology of decoded data.

OBR_NONDT	: No data
OBR_CD39	: Code39 symbology
OBR_NW_7	: NW-7 symbology
OBR_WPCA	: WPC Addon symbology
OBR_WPC	: WPC symbology
OBR_UPEA	: UPC-E Addon symbology
OBR_UPE	: UPC-E symbology
OBR_IDF	: Industrial 2of5 symbology
OBR_ITF	: Interleaved 2of5 symbology
OBR_CD93	: Code93 symbology
OBR_CD128	: Code128 symbology (note 5)
OBR_MSI	: MSI symbology
OBR_IATA	: IATA symbology
OBR_RSS14	: RSS-14 symbology or RSS-14 Stacked symbology. (notes 3 and 4)
OBR_RSSLTD	: RSS Limited symbology. (notes 3 and 4)
OBR_RSSEXP	: RSS Expanded symbology or RSS Expanded Stacked symbology. (notes 3 and 4)

pLength

This parameter is for retrieving the length of the decoded data.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

Notes:

1. To display decoded data retrieved with **OBRGetc** function, they must be converted into Unicode to display in WindowsCE.
2. OBR_NONDT is set in the **pCode** parameter if there is no data in the OBR buffer.
3. IT-600, DT-X7, DT-X11 DT-X30, IT-800, IT-300, and DT-X8 support the symbology.
4. The RSS symbologies change its name to "GS1 DataBar".
RSS-14 → GS1 DataBar Omnidirectional
RSS Limited → GS1 DataBar Limited
RSS Expanded → GS1 DataBar Expanded
RSS-14 Stacked → GS1 DataBar Stacked
RSS Expanded Stacked → GS1 DataBar Expanded Stacked
5. The function retrieves "OBR_CD128" even if reading the GS1-128 symbology (previously named as EAN128 symbology).

4.25 OBRGetStatus

This function retrieves the status of the decoding data storage buffer in the scanner driver.

Calling Sequences

```
[C++]
int OBRGetStatus(
    DWORD *pSize,
    BYTE *pNumber
)
```

```
[Visual Basic]
Public Shared Function OBRGetStatus( _
    ByRef pSize As Int32, _
    ByRef pNumber As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetStatus(
    ref Int32 pSize,
    ref Byte pNumber
);
```

Parameters

pSize

This parameter is for retrieving the number of bytes remained in the buffer (no. of steps remained x maximum length of one data).

pNumber

This parameter is for retrieving the number of steps remained in the buffer.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.26 OBRClearBuff

This function clears the decoding data storage buffer in the scanner driver.

Calling Sequences

```
[C++]  
int OBRClearBuff( )
```

```
[Visual Basic]  
Public Shared Function OBRClearBuff() As Int32
```

```
[C#]  
public static Int32 OBRClearBuff()
```

Parameter

None

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.27 OBRSetBuffType

This function sets up the output mode for decoded data.

Calling Sequences

```
[C++]
int OBRSetBuffType(
    BYTE byType
)
```

```
[Visual Basic]
Public Shared Function OBRSetBuffType( _
    ByVal byType As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetBuffType(
    Byte byType
);
```

Parameters

byType

This parameter is for specifying output mode selecting one of the values listed below.

OBR_BUFOBR	: Output to OBR buffer
OBR_STOFF	: Output as key message to specified window handle
OBR_ASTOFF	: Output to focused location via Clipboard

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

The destination window handle for key message is specified with **OBROpen** function.

4.28 OBRGetBuffType

This function retrieves the decoded data output mode.

Calling Sequences

```
[C++]
int OBRGetBuffType(
    BYTE *pType
)
```

```
[Visual Basic]
Public Shared Function OBRGetBuffType( _
    ByRef pType As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetBuffType(
    ref Byte pType
);
```

Parameters

pType

This is for retrieving the output mode. See **OBRSetBuffType** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.29 OBRSetCheckCounter

This function sets up the number of validations to carry out for decoded data.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetCheckCounter** function.

Calling Sequences

```
[C++]
int OBRSetCheckCounter(
    int nCounter
)
```

```
[Visual Basic]
Public Shared Function OBRSetCheckCounter( _
    ByVal nCounter As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetCheckCounter(
    Int32 nCounter
);
```

Parameters

nCounter

For Laser Scanner;

This parameter is for specifying the number of times in the range of 1 to 9 to carry out validation. The default is 3.

For Linear Imager;

This parameter is for specifying the number of times in the range of 1 to 9 to carry out validation on decoded data. The default setting which is "0" will set the validation to automatic.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.30 OBRGetCheckCounter

This function retrieves the number of times set to carry out validation for decoded data.

Calling Sequences

```
[C++]
int OBRGetCheckCounter(
    int *pCounter
)
```

```
[Visual Basic]
Public Shared Function OBRGetCheckCounter( _
    ByRef pCounter As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetCheckCounter(
    ref Int32 pCounter
);
```

Parameters

pCounter

This parameter is for retrieving the number of times for validation. See **OBRSetCheckCounter** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT	: Unsupported error

4.31 OBRSetScanningTimeout

This function sets up a time period for timeout used for scanning bar code.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetScanningTimeout** function.

Calling Sequences

```
[C++]
int OBRSetScanningTimeout(
    int nTimeout
)
```

```
[Visual Basic]
Public Shared Function OBRSetScanningTimeout( _
    ByVal nTimeout As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetScanningTimeout(
    Int32 nTimeout
);
```

Parameters

nTimeout

This parameter is for specifying a time period in the range of 1 to 9 seconds for timeout. The default is 3 seconds.

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.32 OBRGetScanningTimeout

This function retrieves the time period set for timeout used for scanning bar code.

Calling Sequences

```
[C++]
int OBRGetScanningTimeout(
    int *pTimeout
)
```

```
[Visual Basic]
Public Shared Function OBRGetScanningTimeout( _
    ByRef pTimeout As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetScanningTimeout(
    ref Int32 pTimeout
);
```

Parameters

pTimeout

This parameter is for retrieving the time period. See **OBRSetScanningTimeout** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.33 OBRSetSuffixChar

This function sets up termination control code that is appended to the end of decoded data before output.

Calling Sequences

```
[C++]
int OBRSetSuffixChar(
    BYTE bySuffix
)
```

```
[Visual Basic]
Public Shared Function OBRSetSuffixChar( _
    ByVal bySuffix As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetSuffixChar(
    Byte bySuffix
);
```

Parameters

bySuffix

This parameter is for specifying termination control code selecting one of the values listed below.

OBR_ENDCR	: CR code (0Dh)
OBR_ENDLF	: LF code (0Ah)
OBR_ENDCL	: CR code+LF code (0D0Ah)
OBR_ENDTAB	: HT code (09h)
OBR_ENDNULL	: No termination code.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.34 OBRGetSuffixChar

This function retrieves the termination code at the end of decoded data.

Calling Sequences

```
[C++]
int OBRGetSuffixChar(
    BYTE *pSuffix
)
```

```
[Visual Basic]
Public Shared Function OBRGetSuffixChar( _
    ByRef pSuffix As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetSuffixChar(
    ref Byte pSuffix
);
```

Parameters

pSuffix

This parameter is for retrieving termination control code. See **OBRSetSuffixChar** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.35 OBRSetScanningType

This function sets up scanning mode.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetScanningType** function.

Calling Sequences

```
[C++]
int OBRSetScanningType(
    BYTE byType
)
```

```
[Visual Basic]
Public Shared Function OBRSetScanningType( _
    ByVal byType As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetScanningType(
    Byte byType
);
```

Parameters

byType

This parameter is for specifying scanning mode selecting either of the values listed below.

OBR_TYPE0	: Single-shot scanning/No key release check
OBR_TYPE1	: Continuous scanning/With key release check

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.36 OBRGetScanningType

This function retrieves the scanning mode.

Calling Sequences

```
[C++]
int OBRGetScanningType(
    BYTE *pType
)
```

```
[Visual Basic]
Public Shared Function OBRGetScanningType( _
    ByRef pType As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetScanningType(
    ref Byte pType
);
```

Parameters

pType

This parameter is for retrieving the scanning mode. See **OBRSetScanningType** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.37 OBRSetScanningCounter

This function sets up the number of times to carry out continuous scanning.

Calling Sequences

```
[C++]
int OBRSetScanningCounter(
    int nCounter
)
```

```
[Visual Basic]
Public Shared Function OBRSetScanningCounter( _
    ByVal nCounter As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetScanningCounter(
    Int32 nCounter
);
```

Parameters

nCounter

This parameter is for specifying the number of times in the range of 1 to 9 to carry out continuous scanning. The default is 1 time.

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.38 OBRGetScanningCounter

This function retrieves the number of times set for continuous scanning.

Calling Sequences

```
[C++]
int OBRGetScanningCounter(
    int *pCounter
)
```

```
[Visual Basic]
Public Shared Function OBRGetScanningCounter( _
    ByRef pCounter As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetScanningCounter(
    ref Int32 pCounter
);
```

Parameters

pCounter

This parameter is for retrieving the number of times for continuous scanning. See **OBRSetScanningCounter** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.39 OBRSetMultiStepReading

This function sets up step reading which is described below.

In the Device Emulator, it is not possible to retrieve multiple codes from the I/O Simulator.

Normal Read mode

There is no limit to the number of bar codes that can be scanned once **OBROpen** function has been carried out to open and until **OBRClose** function is carried out to close. When continuous scanning is in operation, a repeated scanning on the same bar code will be avoided if the decoding result is the same with that of the previous decoding result one scanning before.

Multi-step Read mode

Only the specified number of bar codes will be scanned once **OBROpen** function has been carried out to open the scanning driver and until **OBRClose** function is carried out to close. The number of bar codes to be scanned is specified with **OBRSetScanningCounter** function. When continuous scanning is in operation, a repeated scanning on the same bar code will be avoided if the decoding result is the same with that of the previous decoding result one scanning before.

Calling Sequences

```
[C++]
int OBRSetMultiStepReading(
    BYTE byMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetMultiStepReading( _
    ByVal byMode As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetMultiStepReading(
    Byte byMode
);
```

Parameters

byMode

This parameter is for specifying the step reading mode selecting either of the values listed below.

OBR_NORM	: Normal read mode
OBR_DANN	: Multi-step read mode

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.40 OBRGetMultiStepReading

This function retrieves the Multi-step read mode.

Calling Sequences

```
[C++]
int OBRGetMultiStepReading(
    BYTE *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetMultiStepReading( _
    ByRef pMode As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetMultiStepReading(
    ref Byte pMode
);
```

Parameters

pMode

This parameter is for retrieving the Multi-step read mode. See **OBRSetMultiStepReading** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.41 OBRStartScanning

This function turns on the laser and initiates continuous scanning. Scanning in the mode can be stopped in mid-course by carrying out **OBRStopScanning** function. Before calling **OBRStartScanning** function, **OBROpen** function must be called first to open the scanner driver. There is no limit to the number of times for scanning. However, care must be taken, because if the power is turned off in the continuous scanning mode, the scanner will automatically resume in operation when the power is turned on.

Calling Sequences

```
[C++]
int OBRStartScanning(
    DWORD dwTimeout
)
```

```
[Visual Basic]
Public Shared Function OBRStartScanning( _
    ByVal dwTimeout As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRStartScanning(
    Int32 dwTimeout
);
```

Parameters

dwTimeout

This parameter is for specifying a time period for timeout in the range of 1 to 300 seconds or OBR_INFINITE (no timeout).

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
OBR_ERROR_SCANNING	: Mid scan error
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

After scanning one bar code and then no subsequent action is taken within the time period set in the *dwTimeout* parameter, scanning will stop.

4.42 OBRStopScanning

This function stops continuous scanning. If this function is called in mid-course while scanning continues initiated with **OBRStartScanning** function, the scanning will stop. Before calling **OBRStopScanning** function, **OBROpen** function must be called first to open the scanner driver.

Calling Sequences

```
[C++]  
int OBRStopScanning( )
```

```
[Visual Basic]  
Public Shared Function OBRStopScanning() As Int32
```

```
[C#]  
public static Int32 OBRStopScanning()
```

Parameter

None

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.43 OBRSetDecodeLevel

This function sets up the decoding level in standard level or high level.

In the Device Emulator, the function does not perform, but stores the setting value as internal variable. The value stored can be checked with **OBRGetDecodeLevel** function.

Switching decoding level means that if a bar code in poor quality caused by irregular bar ratio is scanned, and its result is in error. The decoding level can be switched to the other to successfully scan such the poor bar code.

Standard level

This level allows scanning with the standard decode logic.

High level

Initially employs the standard decode logic to scan, but its result is in error. In this case, the other decoding level ("High level") implemented in the terminal is used to scan the same bar code. This improves the error rate specifically with EAN, UPC-A, Code39, and Code128 symbologies.

However, with other bar code symbologies, neither it does improve the error rate, nor may reduce miss-reading. It is recommended that the high level only be used if scanning is not possible using the standard level.

Calling Sequences

```
[C++]
int OBRSetDecodeLevel(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetDecodeLevel( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetDecodeLevel(
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying decode level selecting either of the values listed below.

OBR_DECODELEVEL_NORMAL	: Standard level
OBR_DECODELEVEL_HIGH	: High level

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.44 OBRGetDecodeLevel

This function retrieves decoding level.

Calling Sequences

```
[C++]
int OBRGetDecodeLevel(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetDecodeLevel( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetDecodeLevel(
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is for retrieving decoding level. See **OBRSetDecodeLevel** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.45 OBRSetScanningCode

This function sets up "Enable" or "Disable" for scanning each bar code symbology.

Calling Sequences

```
[C++]
int OBRSetScanningCode(
    DWORD dwMode
)
```

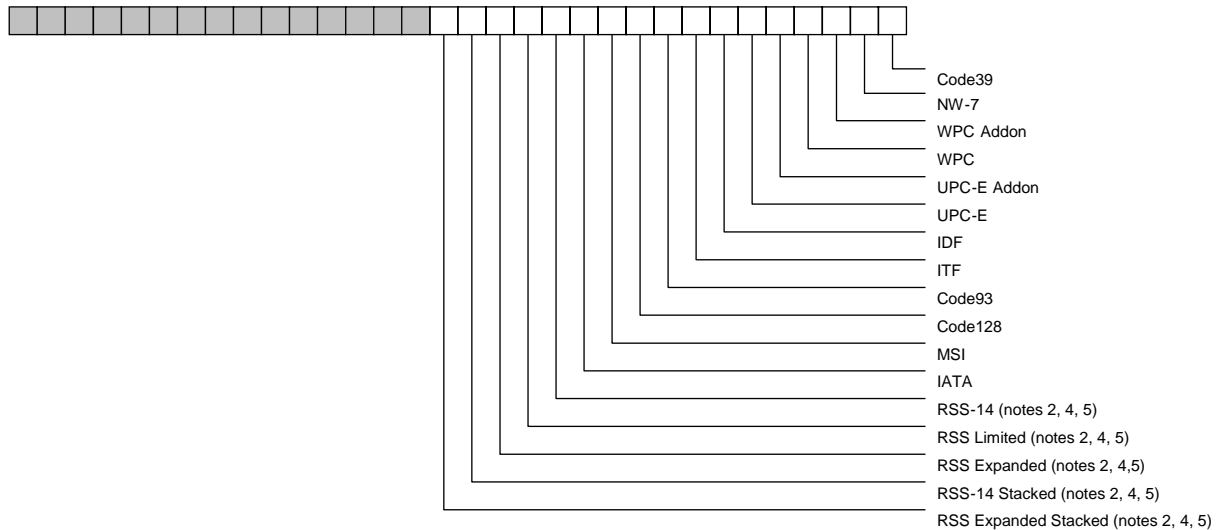
```
[Visual Basic]
Public Shared Function OBRSetScanningCode( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetScanningCode(
    Int32 dwMode
);
```


Parameters

dwMode

This parameter is for specifying "Enable" or "Disable" for scanning each symbology using a bit field below. To set multiple symbologies enabled, a sum of the values in logical OR for specified symbologies is set in this parameter.



OBR_CD39	: Enable Code39 symbology.
OBR_NW_7	: Enable NW-7 symbology.
OBR_WPCA	: Enable WPC Addon symbology.
OBR_WPC	: Enable WPC symbology.
OBR_UPEA	: Enable UPC-E Addon symbology.
OBR_UPE	: Enable UPC-E symbology.
OBR_IDF	: Enable Industrial 2of5 symbology. (note 3)
OBR_ITF	: Enable Interleaved 2of5 symbology.
OBR_CD93	: Enable Code93 symbology.
OBR_CD128	: Enable Code128 symbology. (note 6)
OBR_MSI	: Enable MSI symbology.
OBR_IATA	: Enable IATA symbology. (note 3)
OBR_RSS14	: Enable RSS-14 symbology. (notes 2, 4, 5)
OBR_RSSLTD	: Enable RSS Limited symbology. (notes 2, 4, 5)
OBR_RSSEXP	: Enable RSS Expanded symbology. (notes 2, 4, 5)
OBR_RSS14S	: Enable RSS-14 Stacked symbology enabled. (notes 2, 4, 5)
OBR_RSSEXP	: Enable RSS Expanded stacked symbology enabled. (notes 2, 4, 5)
OBR_ALL	: Enable all the symbologies.

Return Value

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Notes:

1. **OBROpen** function also can be used to specify each "Enable" or "Disable" status for scanning individual symbology. The settings with **OBROpen** function have the priority over the settings with **OBRSetScanningCode** function. This means that the settings on individual symbology performed with **OBRSetScanningCode** function are reset to the settings with **OBROpen** function when **OBROpen** function is called.
2. IT-600, DT-X7, DT-X11, DT-X30, IT-800, IT-300, and DT-X8 support the symbology.
3. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that the settings for reading Industrial 2of5 and IATA symbologies are to be concurrently made enabled. If it is done wrongly, all other settings for reading other symbologies are disabled and a parameter error is returned.
4. Individual settings for reading RSS-14, RSS-14 Stacked, and RSS Expanded, and RSS Expanded Stacked symbologies are not supported. If setting for reading a standard RSS symbology is set enabled, all the corresponded RSS Stacked symbologies are automatically set enabled for reading.
5. The RSS symbologies change its name to "GS1 DataBar".
 - RSS-14 → GS1 DataBar Omnidirectional
 - RSS Limited → GS1 DataBar Limited
 - RSS Expanded → GS1 DataBar Expanded
 - RSS-14 Stacked → GS1 DataBar Stacked
 - RSS Expanded Stacked → GS1 DataBar Expanded Stacked
6. The setting for reading Code128 symbology will make also reading the GS1-128 symbology (previously named as EAN128 symbology) possible.

4.46 OBRGetScanningCode

This function retrieves each "Enable" or "Disable" status for scanning the symbologies.

Calling Sequences

```
[C++]
int OBRGetScanningCode(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetScanningCode( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetScanningCode(
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is for retrieving each "Enable" or "Disable" status for the symbologies. See **OBRSetScanningCode** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.47 OBRSetCode39Option

This function sets up the decode options for Code39 symbology.

Calling Sequences

```
[C++]
int OBRSetCode39Option(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetCode39Option( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetCode39Option(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with Code39 symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable Code39 symbology.
OBR_CODE_DISABLE	: Disable Code39 symbology.
OBR_CODE_IGNORE	: Continue with the current setting.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of Code39 symbology with its length shorter than the value set in this parameter will not be decoded. The default is 2.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of Code39 symbology with its length greater than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X7
52 digits
- DT-X11
47 digits
- IT-600
52 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
48 digits

byOutFormat

This parameter is for specifying output format selecting one of the values listed below.

OBR_39SON	: Output Code39 symbology with start and stop codes.
OBR_39SOFF	: Output Code39 symbology without start and stop codes.
OBR_39ASON	: Output Code39 symbology in Full ASCII and with start and stop codes.
OBR_39ASOFF	: Output Code39 symbology in Full ASCII and without start and stop codes.

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: With check digit calculation

byCheckChar

This parameter is for specifying check character output selecting either of the values listed below.

OBR_CHKKOFF	: Without check character output
OBR_CHKKON	: With check character output

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.48 OBRGetCode39Option

This function retrieves the decode options for Code39 symbology.

Calling Sequences

```
[C++]
int OBRGetCode39Option(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetCode39Option( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetCode39Option(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with Code39 symbology. See **OBRSetCode39Option** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetCode39Option** function for the values to capture.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetCode39Option** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetCode39Option** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit setting. See **OBRSetCode39Option** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character output. See **OBRSetCode39Option** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.49 OBRSetNW7Option

This function sets up the decode options for NW-7 symbology.

Calling Sequences

```
[C++]
int OBRSetNW7Option(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat
)
```

```
[Visual Basic]
Public Shared Function OBRSetNW7Option( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetNW7Option(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat
);
```

Parameters

byEnable

This parameter is for specifying scanning status with NW-7 symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable NW-7 symbology.
OBR_CODE_DISABLE	: Disable NW-7 symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of NW-7 symbology with its length less than the value set in this parameter will not be decoded. The default is 2.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of NW-7 symbology with its length greater than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11
62 digits
- IT-600
63 digits
- DT-X7
63 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
48 digits

byOutFormat

This parameter is for specifying output format selecting either of the values listed below.

- OBR_NWSON : Output NW-7 symbology with start and stop codes.
OBR_NWSOFF : Output NW-7 symbology without start and stop codes.

Return Values

- OBR_OK : Normal end
OBR_PRM : Parameter error
OBR_PON : Open already
OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport : Unsupported error

4.50 OBRSetNW7OptionEx

This function sets up the decode options for NW-7 symbology.

In the Device Emulator, the function returns always FUNCTION_UNSupport.

Calling Sequences

```
[C++]
int OBRSetNW7OptionEx(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetNW7OptionEx( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetNW7OptionEx(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with NW-7 symbology selecting one of the values listed below.

- | | |
|------------------|---------------------------------------|
| OBR_CODE_ENABLE | : Enable NW-7 symbology. |
| OBR_CODE_DISABLE | : Disable NW-7 symbology. |
| OBR_CODE_IGNORE | : Continue with the current settings. |

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of NW-7 symbology with its length shorter than the value set in this parameter will not be decoded. The default is 2.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of NW-7 symbology with its length longer than the value set in this parameter will not be decoded. The default value for IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E is as follows.

- 48 digits

byOutFormat

This parameter is for specifying output format selecting either of the values listed below.

- OBR_NWSON : Output NW-7 symbology with start and stop codes.
- OBR_NWSOFF : Output NW-7 symbology without start and stop codes.

byCheckDigit

This is parameter for specifying check digit selecting either of the values listed below.

- OBR_CHKDOFF : Without check digit calculation
- OBR_CHKDON : With check digit calculation

byCheckChar

This parameter is for specifying check character. Since the check digit calculation with NW-7 symbology is always disabled, specify OBR_CHKKOFF in this parameter.

Return Values

- OBR_OK : Normal end
- OBR_PRM : Parameter error
- OBR_PON : Open already
- OBR_NOT_DEVICE : Scanner driver error
- FUNCTION_UN SUPPORT : Unsupported error

4.51 OBRGetNW7Option

This function retrieves the decode options for NW-7 symbology.

Calling Sequences

```
[C++]
int OBRGetNW7Option(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetNW7Option( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetNW7Option(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with NW-7 symbology. See **OBRSetNW7OptionEx** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetNW7OptionEx** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetNW7OptionEx** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetNW7OptionEx** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetNW7OptionEx** function for the values to retrieve.

pCheckChar

This function is for retrieving the check character output. See **OBRSetNW7OptionEx** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.52 OBRSetWPCAddonOption

This function sets up the decode options for WPC Addon symbology.

Calling Sequences

```
[C++]
int OBRSetWPCAddonOption(
    BYTE byEnable,
    BYTE byCheckDigit
)
```

```
[Visual Basic]
Public Shared Function OBRSetWPCAddonOption( _
    ByVal byEnable As Byte, _
    ByVal byCheckDigit As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetWPCAddonOption(
    Byte byEnable,
    Byte byCheckDigit
);
```

Parameters

byEnable

This parameter is for specifying scanning status with WPC Addon symbology selecting one of the values listed below. See note 1.

OBR_CODE_ENABLE	: Enable WPC Addon symbology.
OBR_CODE_DISABLE	: Disable WPC Addon symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the WPC symbology. See note 2.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: With check digit calculation

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support reading UPC-B Addon symbology.
2. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in *byCheckDigit* parameter.

4.53 OBRSetWPCAddonOptionEx

This function sets up the decode options for WPC Addon symbology.

Calling Sequences

```
[C++]
int OBRSetWPCAddonOptionEx(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetWPCAddonOptionEx( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetWPCAddonOptionEx(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with WPC Addon symbology selecting one of the values listed below. See note 1.

OBR_CODE_ENABLE	: Enable WPC Addon symbology.
OBR_CODE_DISABLE	: Disable WPC Addon symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed to 10 digits and cannot be changed. Always specify 10.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 18 digits and cannot be changed. Always specify 18.

byOutFormat

This parameter is for specifying output format selecting either of the values listed below.

OBR_NUMSYSON	: Output UPC-A symbology with "0" at the forefront.
OBR_NUMSYSOFF	: Output UPC-A symbology without "0" at the forefront.
OBR_WPCGTIN	: Output UPC-A, EAN8, and EAN13 symbologies in GTIN format (14 digits).

byCheckDigit

This is parameter for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the WPC symbology. See note 2.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: With check digit calculation

byCheckChar

This parameter is for specifying check character. Since the check digit calculation with WPC Addon symbology is always enabled, specify OBR_CHKKON in this parameter.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support reading UPC-B Addon symbology.
2. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in ***byCheckDigit*** parameter.

4.54 OBRGetWPCAddonOption

This function retrieves the decode options for WPC Addon symbology.

Calling Sequences

```
[C++]
int OBRGetWPCAddonOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetWPCAddonOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetWPCAddonOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with WPC Addon symbology. See **OBRSetWPCAddonOptionEx** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetWPCAddonOptionEx** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetWPCAddonOptionEx** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetWPCAddonOptionEx** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetWPCAddonOptionEx** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetWPCAddonOptionEx** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.55 OBRSetWPCOption

This function sets up the decode options for WPC symbology.

Calling Sequences

```
[C++]
int OBRSetWPCOption(
    BYTE byEnable,
    BYTE byCheckDigit
)
```

```
[Visual Basic]
Public Shared Function OBRSetWPCOption( _
    ByVal byEnable As Byte, _
    ByVal byCheckDigit As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetWPCOption(
    Byte byEnable,
    Byte byCheckDigit
);
```

Parameters

byEnable

This parameter is for specifying scanning status with WPC symbology selecting one of the values listed below. See note 1.

- OBR_CODE_ENABLE : Enable WPC symbology.
- OBR_CODE_DISABLE : Disable WPC symbology.
- OBR_CODE_IGNORE : Continue with the current settings.

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the WPC symbology. See note 2.

- OBR_CHKDOFF : Without check digit calculation
- OBR_CHKDON : With check digit calculation

Return Values

- OBR_OK : Normal end
- OBR_PRM : Parameter error
- OBR_PON : Open already
- OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
- FUNCTION_UNSUPPORTED : Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support reading UPC-B Addon symbology.
2. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in ***byCheckDigit*** parameter.

4.56 OBRSetWPCOptionEx

This function sets up the decode options for WPC symbology.

Calling Sequences

```
[C++]
int OBRSetWPCOptionEx(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetWPCOptionEx( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetWPCOptionEx(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with WPC symbology selecting one of the values listed below. See note 1.

- | | |
|------------------|---------------------------------------|
| OBR_CODE_ENABLE | : Enable WPC symbology. |
| OBR_CODE_DISABLE | : Disable WPC symbology. |
| OBR_CODE_IGNORE | : Continue with the current settings. |

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed to 8 digits and cannot be changed. Always specify 8.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 13 digits and cannot be changed. Always specify 13.

byOutFormat

This parameter is for specifying output format selecting one of the values listed below.

- OBR_NUMSYSON : Output UPC-A symbology with "0" at the forefront.
- OBR_NUMSYSOFF : Output UPC-A symbology without "0" at the forefront.
- OBR_WPCGTIN : Output UPC-A, EAN8, and EAN13 symbologies in GTIN format (14 digits).

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the WPC Addon symbology. See note 2.

- OBR_CHKDOFF : Without check digit calculation
- OBR_CHKDON : With check digit calculation

byCheckChar

This parameter is for specifying check character. Since the check digit calculation with WPC symbology is always enabled, OBR_CHKKON must be specified in this parameter.

Return Values

- OBR_OK : Normal end
- OBR_PRM : Parameter error
- OBR_PON : Open already
- OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
- FUNCTION_UNSUPPORTED : Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support reading UPC-B Addon symbology.
2. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in ***byCheckDigit*** parameter.

4.57 OBRGetWPCOption

This function retrieves the decode options for WPC symbology.

Calling Sequences

```
[C++]
int OBRGetWPCOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetWPCOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetWPCOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with WPC symbology. See **OBRSetWPCOptionEx** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetWPCOptionEx** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetWPCTOptionEx** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetWPCTOptionEx** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetWPCTOptionEx** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetWPCTOptionEx** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.58 OBRSetUPCEAddonOption

This function sets up the decode options for UPC-E Addon symbology.

Calling Sequences

```
[C++]
int OBRSetUPCEAddonOption(
    BYTE byEnable,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetUPCEAddonOption( _
    ByVal byEnable As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetUPCEAddonOption(
    Byte byEnable,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with UPC-E Addon symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable UPC-E Addon symbology.
OBR_CODE_DISABLE	: Disable UPC-E Addon symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the UPC-E symbology. See note.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: With check digit calculation

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

OBR_CHKKOFF	: Without check character output
OBR_CHKKON	: With check character output

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in ***byCheckDigit*** parameter.

4.59 OBRSetUPCEAddonOptionEx

This function sets up the decode options for UPC-E Addon symbology.

Calling Sequences

```
[C++]
int OBRSetUPCEAddonOptionEx(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetUPCEAddonOptionEx( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetUPCEAddonOptionEx(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with UPC-E Addon symbology selecting one of the values below.

OBR_CODE_ENABLE	: Enable UPC-E Addon symbology.
OBR_CODE_DISABLE	: Disable UPC-E Addon symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed to 9 digits and cannot be changed. Always specify 9.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 12 digits and cannot be changed. Always specify 12.

byOutFormat

This parameter is for specifying output format selecting either of the values listed below.

- OBR_NUMSYSON : Output UPC-E Addon symbology with "0" at the forefront.
- OBR_NUMSYSOFF : Output UPC-E Addon symbology without "0" at the forefront.

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the UPC-E symbology. See note.

- OBR_CHKDOFF : Without check digit calculation
- OBR_CHKDON : With check digit calculation

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

- OBR_CHKKOFF : Without check character output
- OBR_CHKKON : With check character output

Return Values

- OBR_OK : Normal end
- OBR_PRM : Parameter error
- OBR_PON : Open already
- OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
- FUNCTION_UNSUPPORTED : Unsupported error

Note:

IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in ***byCheckDigit*** parameter.

4.60 OBRGetUPCEAddonOption

This function retrieves the decode options for UPC-E Addon symbology.

Calling Sequences

```
[C++]
int OBRGetUPCEAddonOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetUPCEAddonOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetUPCEAddonOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with UPC-E Addon symbology. See **OBRSetUPCEAddonOptionEx** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetUPCEAddonOptionEx** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetUPCEAddonOptionEx** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetUPCEAddonOptionEx** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetUPCEAddonOptionEx** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetUPCEAddonOptionEx** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error
FUNCTION_UNSupport	: Unsupported error

4.61 OBRSetUPCEOption

This function sets up the decode options for UPC-E symbology.

Calling Sequences

```
[C++]
int OBRSetUPCEOption(
    BYTE byEnable,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetUPCEOption( _
    ByVal byEnable As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetUPCEOption(
    Byte byEnable,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with UPC-E symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable UPC-E symbology.
OBR_CODE_DISABLE	: Disable UPC-E symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the UPC-E Addon symbology. See note.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: With check digit calculation

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

OBR_CHKKOFF	: Without check character output
OBR_CHKKON	: With check character output

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT	: Unsupported error

Note:

IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in **byCheckDigit** parameter.

4.62 OBRSetUPCEOptionEx

This function sets up the decode options for UPC-E symbology.

Calling Sequences

```
[C++]
int OBRSetUPCEOptionEx(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetUPCEOptionEx( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetUPCEOptionEx(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with UPC-E symbology selecting one of the values listed below.

- | | |
|------------------|---------------------------------------|
| OBR_CODE_ENABLE | : Enable UPC-E symbology. |
| OBR_CODE_DISABLE | : Disable UPC-E symbology. |
| OBR_CODE_IGNORE | : Continue with the current settings. |

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed 7 digits and cannot be changed. Always specify 7.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 7 digits and cannot be changed. Always specify 7.

byOutFormat

This parameter is for specify output format selecting one of the values listed below.

- OBR_NUMSYSON : Output UPC-E symbology with "0" at the forefront.
- OBR_NUMSYSOFF : Output UPC-E symbology without "0" at the forefront.
- OBR_UPCEGTIN : Output UPC-E symbology in GTIN format (14 digits).

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below. See note. To set "OBR_CHKDOFF" in the parameter, be sure also to set the same value for the UPC-E Addon. See note.

- OBR_CHKDOFF : Without check digit calculation
- OBR_CHKDON : With check digit calculation

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

- OBR_CHKKOFF : Without check character output
- OBR_CHKKON : With check character output

Return Values

- OBR_OK : Normal end
- OBR_PRM : Parameter error
- OBR_PON : Open already
- OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
- FUNCTION_UN SUPPORT : Unsupported error

Note:

IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in **byCheckDigit** parameter.

4.63 OBRGetUPCEOption

This retrieves the decode options for UPC-E symbology.

Calling Sequences

```
[C++]
int OBRGetUPCEOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetUPCEOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetUPCEOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with UPC-E symbology. See **OBRSetUPCEOptionEx** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetUPCEOptionEx** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetUPCEOptionEx** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetUPCEOptionEx** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetUPCEOptionEx** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetUPCEOptionEx** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.64 OBRSetIDFOption

This function sets up the decode options for Industrial 2of5 symbology.

Calling Sequences

```
[C++]
int OBRSetIDFOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetIDFOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetIDFOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with Industrial 2of5 symbology selecting one of the values listed below. See note.

OBR_CODE_ENABLE	: Enable Industrial 2of5 symbology.
OBR_CODE_DISABLE	: Disable Industrial 2of5 symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of Industrial 2of5 symbology with its length less than the value set in this parameter will not be decoded. The default is 2.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of Industrial 2of5 symbology with its length longer than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11
55 digits
- IT-600
67 digits
- DT-X7
67 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
50 digits

byCheckDigit

This parameter is for specifying designating the check digit selecting either of the values listed below.

- OBR_CHKDOFF : Without check digit calculation
OBR_CHKDON : With check digit calculation

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

- OBR_CHKKOFF : Without check character output
OBR_CHKKON : With check character output

Return Values

- OBR_OK : Normal end
OBR_PRN : Parameter error
OBR_PON : Open already
OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED : Unsupported error

Note:

IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that the settings for reading Industrial 2of5 and IATA symbologies are to be concurrently made enabled. If it is done wrongly, all other settings for reading other symbologies are disabled and a parameter error is returned.

4.65 OBRGetIDFOption

This function retrieves the decode options for Industrial 2of5 symbology.

Calling Sequences

```
[C++]
int OBRGetIDFOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetIDFOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetIDFOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with IDF symbology. See **OBRSetIDFOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetIDFOption** function for values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetIDFOption** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetIDFOption** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetIDFOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetIDFOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.66 OBRSetITFOption

This function sets up the decode options for Interleaved 2of5 symbology.

Calling Sequences

```
[C++]
int OBRSetITFOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetITFOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetITFOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with Interleaved 2of5 symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable Interleaved 2of5 symbology.
OBR_CODE_DISABLE	: Disable Interleaved 2of5 symbology.
OBR_CODE_IGNORE	: Continue with current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of Interleaved 2of5 symbology with its length less than the value set in this parameter will not be decoded. The default is 4.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of Interleaved 2of5 symbology with its length longer than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11
92 digits
- IT-600
94 digits
- DT-X7
94 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
50 digits

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below.

OBR_CHKDOFF : Without check digit calculation
OBR_CHKDON : With check digit calculation

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

OBR_CHKKOFF : Without check character output
OBR_CHKKON : With check character output

Return Value

OBR_OK : Normal end
OBR_PRM : Parameter error
OBR_PON : Open already
OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT : Unsupported error

4.67 OBRGetITFOption

This function retrieves the decode options for Interleaved 2of5 symbology.

Calling Sequences

```
[C++]
int OBRGetITFOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetITFOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetITFOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with ITF symbology. See **OBRSetITFOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetITFOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetITFOption** function for the values to retrieve.

pOutFormat

This parameter returns always NULL.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetITFOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetITFOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.68 OBRSetCode93Option

This function sets up the decode options for Code93 symbology.

Calling Sequences

```
[C++]
int OBRSetCode93Option(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byCheckDigit
)
```

```
[Visual Basic]
Public Shared Function OBRSetCode93Option( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byCheckDigit As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetCode93Option(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byCheckDigit
);
```

Parameters

byEnable

This parameter is for specifying scanning status with Code93 symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable Code93 symbology.
OBR_CODE_DISABLE	: Disable Code93 symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of Code93 symbology with its length less than the value set in this parameter will not be decoded. The default is 3.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of Code93 symbology with its length longer than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11
77 digits
- IT-600
70 digits
- DT-X7
70 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
50 digits

byCheckDigit

This parameter is for specifying check digit selecting either of the values listed below.

- OBR_CHKDOFF : Without check digit calculation
OBR_CHKDON : With check digit calculation

Return Values

- OBR_OK : Normal end
OBR_PRM : Parameter error
OBR_PON : Open already
OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT : Unsupported error

4.69 OBRGetCode93Option

This function retrieves the decode options for Code93 symbology.

Calling Sequences

```
[C++]
int OBRGetCode93Option(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetCode93Option( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetCode93Option(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with Code93 symbology. See **OBRSetCode93Option** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetCode93Option** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetCode93Option** function for the values to retrieve.

pOutFormat

This parameter returns always NULL.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetCode93Option** function for the values to retrieve.

pCheckChar

This parameter returns always NULL.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.70 OBRSetCode128Option

This function sets up the decode options for Code128 symbology.

Calling Sequences

```
[C++]
int OBRSetCode128Option(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit
)
```

```
[Visual Basic]
Public Shared Function OBRSetCode128Option( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetCode128Option(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit
);
```

Parameters

byEnable

This parameter is for specifying scanning status with Code128 symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable Code128 symbology.
OBR_CODE_DISABLE	: Disable Code128 symbology.
OBR_CODE_IGNORE	: Continue the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of Code128 symbology with its length less than the value set in this parameter will not be decoded. The default is 2.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of Code128 symbology with its length longer than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11 and IT-600
98 digits
- DT-X7
98 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
50 digits

byOutFormat

This parameter is for specifying output format selecting one of the values listed below.

OBR_128AON	: Output EAN-128 after ASCII conversion
OBR_128AOFF	: Output EAN-128 without ASCII conversion. See note 1.
OBR_128EAN	: Output EAN-128 only.
OBR_128AIM	: Output EAN-128 only with ID code.
OBR_128FNC	: Convert EAN-128 only from Fnc to GS and then output.

Besides the output formats above, the following values can be selected in the OR calculation for the output format.

OBR_128AIML	: Output EAN-128 with ID code attached.
OBR_128FNCL	: Convert EAN-128 from Fnc to GS and then output.
OBR_128CAOF	: Output Code128 without ASCII conversion. See note 1.
OBR_128EAOF	: Output EAN-128 without ASCII conversion. See note 1.

IT-600 with either its OS Ver. 5044 and Service Pack 1.02, or its OS Ver. 5069, or a later version being preinstalled can support the following output formats.

OBR_128FNC2	: Output Code128 or EAN-128 with FNC2 combined. See notes 2 and 5.
OBR_128FNC4	: Output Code128 or EAN-128 with expanded FNC4 and ASCII conversion.

byCheckDigit

This is for specifying check digit selecting either of the values listed below. See note 3.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: With check digit calculation

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT	: Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that Code128 and EAN-128 symbologies without ASCII conversion are output. If one of the values with "Without ASCII conversion" is specified in **byOutputFormat** parameter, Specify always set as with ASCII conversion.
2. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that "Output Code128 or EAN-128 with FNC2 combined" is disabled in **byOutFormat** parameter. It is always set enabled.
3. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)". Specify always "OBR_CHKDON (With check digit calculation)" in **byCheckDigit** parameter.
4. The maximum number of digits for both CodeA and CodeB symbologies are 64 and 98 respectively for CodeC symbology.
5. EAN128 changes its symbology name to GS1-128.
6. Code128 should be set as with check digit. If you set this setting without check digit by special reason, please check scanning operation carefully because there is possibility to mis-reading by printed quality or some condition.

4.71 OBRGetCode128Option

This function retrieves the decode options for Code128 symbology.

Calling Sequences

```
[C++]
int OBRGetCode128Option(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetCode128Option( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetCode128Option(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with Code128 symbology. See **OBRSetCode128Option** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetCode128Option** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetCode128Option** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetCode128Option** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetCode128Option** function for the values to retrieve.

pCheckChar

This parameter returns always NULL.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.72 OBRSetMSIOption

This function sets up the decode options for MSI symbology.

Calling Sequences

```
[C++]
int OBRSetMSIOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetMSIOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetMSIOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with MSI symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable MSI symbology.
OBR_CODE_DISABLE	: Disable MSI symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of MSI symbology with its length less than the value set in this parameter will not be decoded. The default is 1.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of MSI symbology with its length longer than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11
67 digits
- IT-600
57 digits
- DT-X7
57 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
50 digits

byCheckDigit

This parameter is for specifying check digit selecting one of the values listed below. See note.

OBR_CHKDOFF	: Without check digit calculation
OBR_CHKDON	: Check digit 1 digit mod10
OBR_CDMSEV	: Check digit 2 digits 1st: mod11 2nd: mod10
OBR_CDMSTN	: Check digit 2 digits 1st: mod10 2nd: mod10

byCheckChar

This parameter is for specifying check character selecting either of the values listed below.

OBR_CHKKOFF	: Without check character output
OBR_CHKKON	: With check character output

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that "OBR_CHKDOFF" and "OBR_CDMSEV" values are set in ***byCheckDigit*** parameter. If one of the values is set wrongly, other settings become invalid and an error is returned.

4.73 OBRGetMSIOption

This function retrieves the decode options for MSI symbology.

Calling Sequences

```
[C++]
int OBRGetMSIOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetMSIOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetMSIOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with MSI symbology. See **OBRSetMSIOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetMSIOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetMSIOption** function for the values to retrieve.

pOutFormat

This parameter returns always NULL.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetMSIOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetMSIOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.74 OBRSetIATAOption

This function sets up the decode options for IATA symbology.

Calling Sequences

```
[C++]
int OBRSetIATAOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byCheckDigit
)
```

```
[Visual Basic]
Public Shared Function OBRSetIATAOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byCheckDigit As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetIATAOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byCheckDigit
);
```

Parameters

byEnable

This parameter is for specifying scanning status with IATA symbology selecting one of the values below. See note 1.

OBR_CODE_ENABLE	: Enable IATA symbology.
OBR_CODE_DISABLE	: Disable IATA symbology.
OBR_CODE_IGNORE	: Continue with current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of IATA symbology with its length less than the value set in this parameter will not be decoded. The default is 4.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of IATA symbology with its length longer than the value set in this parameter will not be decoded. Each default of the respective models is as follows.

- DT-X11
44 digits
- IT-600
65 digits
- DT-X7
65 digits
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E
50 digits
- DT-X30, IT-800, IT-300, and DT-X8
65 digits

byCheckDigit

This parameter is for specifying check digit selecting one of the values listed below.

- | | |
|-------------|---|
| OBR_CHKDOFF | : Without check digit calculation (no. of readable digits; 1 to 40) |
| OBR_CHKDON | : With check digit calculation excepting the last character (no. of readable digits; 2 to 40) |
| OBR_CDMSEV | : With check digit calculation for coupon number and numeric section (no. of readable digits; 15 to 17) |
| OBR_CDMSTN | : With check digit calculation for numeric section (no. of readable digits; 15 to 17) |
| OBR_CHKDON4 | : With check digit calculation using mod10. |

Return Values

- | | |
|----------------------|---|
| OBR_OK | : Normal end |
| OBR_PRN | : Parameter error |
| OBR_PON | : Open already |
| OBR_NOT_DEVICE | : Scanner driver error. In the Device Emulator, this value is not returned. |
| FUNCTION_UNSUPPORTED | : Unsupported error |

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that settings for reading both Industrial 2of5 and IATA symbologies are set enabled concurrently. If the settings are tried, nothing will be set and an error is returned.
2. If "OBR_CDMSEV" or "OBR_CDMSTN" is set in ***byCheckDigit*** parameter for reading IATA symbology, the maximum number of digits is 17.

4.75 OBRGetIATAOption

This function retrieves the decode options for IATA symbology.

Calling Sequences

```
[C++]
int OBRGetIATAOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetIATAOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetIATAOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with IATA symbology. See **OBRSetIATAOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetIATAOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetIATAOption** function for the values to retrieve.

pOutFormat

This parameter returns always NULL.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetIATAOption** function for the values to retrieve.

pCheckChar

This parameter returns always NULL.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.76 OBRSetRSS14Option

This function sets up the decode options for RSS-14 symbology.

Calling Sequences

```
[C++]
int OBRSetRSS14Option(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetRSS14Option( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetRSS14Option(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with RSS-14 symbology selecting one of the values listed below. See note 1.

OBR_CODE_ENABLE	: Enable RSS-14 symbology.
OBR_CODE_DISABLE	: Disable RSS-14 symbology.
OBR_CODE_IGNORE	: Continue with current settings.

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed to 14 digits and cannot be changed. Always specify 14.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 14 digits and cannot be changed. Always specify 14.

byOutFormat

This parameter is for specifying output format selecting either of the values listed below.

OBR_RS14NORM	: Output RSS-14 symbology in standard format.
OBR_RS14AIOF	: Output RSS-14 symbology without application identifier and "01" at the forefront.

byCheckDigit

This parameter is for specifying check digit. Since the check digit with RSS-14 symbology is kept enabled at all times, always specify OBR_CHKDON in this parameter.

byCheckChar

This parameter is for specifying check character output. Since the check character output with RSS-14 symbology is kept enabled at all times, always specify OBR_CHKKON in this parameter.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT	: Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that setting for reading RSS-14 symbology only is individually set enabled. If it is set enabled, setting for reading RSS-14 Stacked symbology is also set enabled automatically.
2. RSS-14 symbology changes its symbology name to "GS1 DataBar Omnidirectional".

4.77 OBRGetRSS14Option

This function retrieves the decode options for RSS-14 symbology.

Calling Sequences

```
[C++]
int OBRGetRSS14Option(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetRSS14Option( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetRSS14Option(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with RSS-14 symbology. See **OBRSetRSS14Option** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetRSS14Option** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetRSS14Option** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetRSS14Option** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetRSS14Option** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character output. See **OBRSetRSS14Option** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

RSS-14 symbology changes its name to "GS1 DataBar Omnidirectional".

4.78 OBRSetRSSLimitedOption

This function sets up the decode options for RSS Limited symbology.

Calling Sequences

```
[C++]
int OBRSetRSSLimitedOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetRSSLimitedOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetRSSLimitedOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with RSS Limited symbology selecting one of the values listed below.

OBR_CODE_ENABLE	: Enable RSS Limited symbology.
OBR_CODE_DISABLE	: Disable RSS Limited symbology.
OBR_CODE_IGNORE	: Continue with the current settings.

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed to 14 digits and cannot be changed. Always specify 14.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 14 digits and cannot be changed. Always specify NULL.

byOutFormat

This parameter is for specify output format selecting either of the values listed below.

- OBR_RSLMNORM : Output RSS Limited symbology in standard format.
- OBR_RSLMAIOF : Output RSS Limited symbology without application identifier and "01" at the forefront.

byCheckDigit

This parameter is for specifying check digit. Since the check digit calculation with RSS Limited symbology is kept enabled at all times, always specify OBR_CHKDON.

byCheckChar

This parameter is for specifying check character. Since the check character output with RSS Limited symbology is kept enabled at all times, always specify OBR_CHKKON.

Return Value

- OBR_OK : Normal end
- OBR_PRM : Parameter error
- OBR_PON : Open already
- OBR_NOT_DEVICE : Scanner driver error. In the Device Emulator, this value is not returned.
- FUNCTION_UNSupport : Unsupported error

Note:

RSS Limited symbology changes its symbology name to "GS1 DataBar Limited".

4.79 OBRGetRSSLimitedOption

This function retrieves the decode options for RSS Limited symbology.

Calling Sequences

```
[C++]
int OBRGetRSSLimitedOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetRSSLimitedOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetRSSLimitedOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with RSS Limited symbology. See **OBRSetRSSLimitedOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetRSSLimitedOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetRSSLimitedOption** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetRSSLimitedOption** function for the values to retrieve.

pCheckDigit

This parameter is for capturing the check digit. See **OBRSetRSSLimitedOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetRSSLimitedOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

Note:

RSS Limited symbology changes its symbology name to "GS1 DataBar Limited".

4.80 OBRSetRSSEExpandedOption

This function sets up the decode options for RSS Expanded symbology.

Calling Sequences

```
[C++]
int OBRSetRSSEExpandedOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetRSSEExpandedOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetRSSEExpandedOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with RSS Expanded symbology selecting one of the values listed below. See note 1.

OBR_CODE_ENABLE	: Enable RSS Expanded symbology.
OBR_CODE_DISABLE	: Disable RSS Expanded symbology.
OBR_CODE_IGNORE	: Continue with current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of RSS Expanded symbology with its length less than the value set in this parameter will not be decoded. The default is 1.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of RSS Expanded symbology with its length greater than the value set in this parameter will not be decoded. The default is 74.

byOutFormat

This parameter is for specifying output format. Since no output format setting with RSS Expanded symbology is required, always specify "NULL".

byCheckDigit

This parameter is for specifying check digit. Since the check digit calculation with RSS Expanded symbology is kept enabled at all times, always specify OBR_CHKDON.

byCheckChar

This parameter is for specifying check character. Since the check character output with RSS Expanded symbology is kept enabled at all times, always specify OBR_CHKKON.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UN SUPPORT	: Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that setting for reading RSS Expanded symbology only is individually set enabled. If it is set enabled, setting for reading RSS Expanded Stacked symbology is also set enabled automatically.
2. RSS Expanded symbology changes its symbology name to "GS1 DataBar Expanded".

4.81 OBRGetRSSExpandedOption

This function retrieves the decode options for RSS Expanded symbology.

Calling Sequences

```
[C++]
int OBRGetRSSExpandedOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetRSSExpandedOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetRSSExpandedOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with RSS Expanded symbology. See **OBRSetRSSExpandedOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetRSSExpandedOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetRSSEExpandedOption** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetRSSEExpandedOption** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetRSSEExpandedOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetRSSEExpandedOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

Note:

RSS Expanded symbology changes its symbology name to "GS1 DataBar Expanded".

4.82 OBRSetRSS14StackedOption

This function sets up the decode options for RSS-14 Stacked symbology.

In the Device Emulator, the function returns always FUNCTION_UNSUPORT.

Calling Sequences

```
[C++]
int OBRSetRSS14StackedOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetRSS14StackedOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetRSS14StackedOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with RSS-14 Stacked symbology selecting one of the values listed below. See note 1.

- | | |
|------------------|-------------------------------------|
| OBR_CODE_ENABLE | : Enable RSS-14 Stacked symbology. |
| OBR_CODE_DISABLE | : Disable RSS-14 Stacked symbology. |
| OBR_CODE_IGNORE | : Continue with current settings. |

byMinLength

This parameter is for specifying the minimum number of digits. It is fixed to 14 digits and cannot be changed. Always specify 14.

byMaxLength

This parameter is for specifying the maximum number of digits. It is fixed to 14 digits and cannot be changed. Always specify 14.

byOutFormat

This parameter is for specifying output format selecting either of the values listed below.

- | | |
|--------------|--|
| OBR_RS14NORM | : Output RSS14 Stacked symbology in standard format. |
| OBR_RS14AIOF | : Output RSS14 Stacked symbology without application identifier and "01" at the forefront. |

byCheckDigit

This parameter is for specifying check digit. Since the check digit with RSS-14 Stacked symbology is kept enabled at all times, always specify OBR_CHKDON in this parameter.

byCheckChar

This parameter is for specifying check character output. Since the check character output with RSS-14 Stacked symbology is kept enabled at all times, always specify OBR_CHKKON in this parameter.

Return Values

- | | |
|---------------------|------------------------|
| OBR_OK | : Normal end |
| OBR_PRM | : Parameter error |
| OBR_PON | : Open already |
| OBR_NOT_DEVICE | : Scanner driver error |
| FUNCTION_UN SUPPORT | : Unsupported error |

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that setting for reading RSS-14 Stacked symbology only is individually set enabled. If it is set enabled, setting for reading RSS-14 symbology is also set enabled automatically.
2. RSS-14 Stacked symbology changes its symbology name to "GS1 DataBar Stacked".

4.83 OBRGetRSS14StackedOption

This function retrieves the decode options for RSS-14 Stacked symbology.

In the Device Emulator, the function returns always FUNCTION_UN SUPPORT.

Calling Sequences

```
[C++]
int OBRGetRSS14StackedOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetRSS14StackedOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetRSS14StackedOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving scanning status with RSS-14 Stacked symbology. See **OBRSetRSS14StackedOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetRSS14StackedOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetRSS14StackedOption** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetRSS14StackedOption** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetRSS14StackedOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character output. See **OBRSetRSS14StackedOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error
FUNCTION_UNSupport	: Unsupported error

Note:

RSS-14 Stacked symbology changes its symbology name to "GS1 DataBar Stacked".

4.84 OBRSetRSSExpandedStackedOption

This function sets up the decode options for RSS Expanded Stacked symbology.

In the Device Emulator, the function returns always FUNCTION_UNSUPPORTED.

Calling Sequences

```
[C++]
int OBRSetRSSExpandedStackedOption(
    BYTE byEnable,
    BYTE byMinLength,
    BYTE byMaxLength,
    BYTE byOutFormat,
    BYTE byCheckDigit,
    BYTE byCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRSetRSSExpandedStackedOption( _
    ByVal byEnable As Byte, _
    ByVal byMinLength As Byte, _
    ByVal byMaxLength As Byte, _
    ByVal byOutFormat As Byte, _
    ByVal byCheckDigit As Byte, _
    ByVal byCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRSetRSSExpandedStackedOption(
    Byte byEnable,
    Byte byMinLength,
    Byte byMaxLength,
    Byte byOutFormat,
    Byte byCheckDigit,
    Byte byCheckChar
);
```

Parameters

byEnable

This parameter is for specifying scanning status with RSS Expanded Stacked symbology selecting one of the values listed below. See note 1.

- OBR_CODE_ENABLE : Enable RSS Expanded Stacked symbology.
- OBR_CODE_DISABLE : Disable RSS Expanded Stacked symbology.
- OBR_CODE_IGNORE : Continue with current settings.

byMinLength

This parameter is for specifying the minimum number of digits. Any bar code of RSS Expanded Stacked symbology with its length less than the value set in this parameter will not be decoded. The default is 1.

byMaxLength

This parameter is for specifying the maximum number of digits. Any bar code of RSS Expanded Stacked symbology with its length greater than the value set in this parameter will not be decoded. The default is 74.

byOutFormat

This parameter is for specifying output format. Since no output format setting with RSS Expanded Stacked symbology is required, always specify "NULL".

byCheckDigit

This parameter is for specifying check digit. Since the check digit calculation with RSS Expanded Stacked symbology is kept enabled at all times, always specify OBR_CHKDON.

byCheckChar

This parameter is for specifying check character. Since the check character output with RSS Expanded Stacked symbology is kept enabled at all times, always specify OBR_CHKKON.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error
FUNCTION_UN SUPPORT	: Unsupported error

Notes:

1. IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support that setting for reading RSS Expanded Stacked symbology only is individually set enabled. If it is set enabled, setting for reading RSS Expanded symbology is also set enabled automatically.
2. RSS Expanded Stacked symbology changes its symbology name to "GS1 DataBar Expanded Stacked".

4.85 OBRGetRSSExpandedStackedOption

This function retrieves the decode options for RSS Expanded Stacked symbology.

In the Device Emulator, the function returns always FUNCTION_UN SUPPORT.

Calling Sequences

```
[C++]
int OBRGetRSSExpandedStackedOption(
    BOOL *pEnable,
    BYTE *pMinLength,
    BYTE *pMaxLength,
    BYTE *pOutFormat,
    BYTE *pCheckDigit,
    BYTE *pCheckChar
)
```

```
[Visual Basic]
Public Shared Function OBRGetRSSExpandedStackedOption( _
    ByRef pEnable As Boolean, _
    ByRef pMinLength As Byte, _
    ByRef pMaxLength As Byte, _
    ByRef pOutFormat As Byte, _
    ByRef pCheckDigit As Byte, _
    ByRef pCheckChar As Byte _
) As Int32
```

```
[C#]
public static Int32 OBRGetRSSExpandedStackedOption(
    ref Boolean pEnable,
    ref Byte pMinLength,
    ref Byte pMaxLength,
    ref Byte pOutFormat,
    ref Byte pCheckDigit,
    ref Byte pCheckChar
);
```

Parameters

pEnable

This parameter is for retrieving the scanning status with RSS Expanded Stacked symbology. See **OBRSetRSSExpandedStackedOption** function for the values to retrieve.

pMinLength

This parameter is for retrieving the minimum number of digits. See **OBRSetRSSExpandedStackedOption** function for the values to retrieve.

pMaxLength

This parameter is for retrieving the maximum number of digits. See **OBRSetRSSExpandedStackedOption** function for the values to retrieve.

pOutFormat

This parameter is for retrieving the output format. See **OBRSetRSSExpandedStackedOption** function for the values to retrieve.

pCheckDigit

This parameter is for retrieving the check digit. See **OBRSetRSSExpandedStackedOption** function for the values to retrieve.

pCheckChar

This parameter is for retrieving the check character. See **OBRSetRSSExpandedStackedOption** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error
FUNCTION_UNSupport	: Unsupported error

Note:

RSS Expanded Stacked symbology changes its name to "GS1 DataBar Expanded Stacked".

4.86 OBRSaveConfigFile

This function saves all the settings for the laser scanner driver in the configuration file.

In the Device Emulator, the function saves all the settings for the laser scanner driver in \FlashDisk\System Settings\OBRDRV.ini.

Calling Sequences

```
[C++]  
int OBRSaveConfigFile( )
```

```
[Visual Basic]  
Public Shared Function OBRSaveConfigFile() As Int32
```

```
[C#]  
public static Int32 OBRSaveConfigFile()
```

Parameter

None

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

The name of configuration file is "OBRDRV.ini", and the file is saved at the respective paths below.

- For DT-X30
 \Documents and Settings\System Settings\OBRDRV.ini

See "Configuration File Format" for the configuration file.

4.87 OBRLoadConfigFile

This function changes all the default settings in the configuration file and then loads them to the laser scanner driver. The default settings can be changed with **OBRLoadConfigFile** function. However, to reflect the new settings in the file to the scanner driver, be sure to call **OBRSetDefaultSymbology** function after **OBRLoadConfigFile** function has been called.

Calling Sequences

```
[C++]  
int OBRLoadConfigFile( )
```

```
[Visual Basic]  
Public Shared Function OBRLoadConfigFile() As Int32
```

```
[C#]  
public static Int32 OBRLoadConfigFile()
```

Parameter

None

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error
FUNCTION_UNSupport	: Unsupported error

Note:

See "Configuration File Format" for the configuration file.

4.88 OBRSetFilter

This function sets up the noise filter mode.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetFilter** function.

Calling Sequences

```
[C++]
int OBRSetFilter(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetFilter( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetFilter(
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying noise filter mode selecting one of the values listed below.

OBR_NOFILTER	: Disable noise filter.
OBR_SOFTFILTER	: Enable software filter.
OBR_HARDFILTER	: Enable hardware filter.
OBR_SOFTHARDFILTER	: Enable software and hardware filters.

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.89 OBRGetFilter

This function retrieves the noise filter mode.

Calling Sequences

```
[C++]
int OBRGetFilter(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetFilter( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetFilter(
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is for retrieving the noise filter mode. See **OBRSetFilter** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error
OBR_PRM	: Argument data error
OBR_PON	: Open already
OBR_NOT_DEVICE_DECODE	: Decode (the decoder section) failed to reside. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.90 OBRSetFilterOnTimer

This function sets up a time span between when decoding bar code data starts and until when the noise filter begins to activate.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetFilterOnTimer** function.

Calling Sequences

```
[C++]
int OBRSetFilterOnTimer(
    int nTime
)
```

```
[Visual Basic]
Public Shared Function OBRSetFilterOnTimer( _
    ByVal nTime As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetFilterOnTimer(
    Int32 nTime
);
```

Parameters

nTime

This parameter is for specifying a time span in the range of 1 to 8 seconds.

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_PRM	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSupport	: Unsupported error

4.91 OBRGetFilterOnTimer

This function retrieves the time span between when decoding bar code data starts and until when the noise filter begins to activate.

Calling Sequences

```
[C++]
int OBRGetFilterOnTimer(
    int *pTime
)
```

```
[Visual Basic]
Public Shared Function OBRGetFilterOnTimer( _
    ByRef pTime As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetFilterOnTimer(
    ref Int32 pTime
);
```

Parameters

pTime

This parameter is for retrieving the time span set for the noise filter. See **OBRSetFilterOnTimer** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_PON	: Open already
OBR_PRM	: Parameter error
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.92 OBRSetGainControl

This function sets up gain control.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetGainControl** function.

Calling Sequences

```
[C++]
int OBRSetGainControl(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetGainControl ( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetGainControl (
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying gain control selecting one of the values listed below.

OBR_GAIN_MODE0	: Gain mode 0
OBR_GAIN_MODE1	: Gain mode 1
OBR_GAIN_MODE2	: Gain mode 2
OBR_GAIN_MODE3	: Gain mode 3
OBR_GAIN_AUTO	: Gain auto (For DT-X7, IT-800, IT-300, and DT-X8)

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

Setting up the scan width control function (OBRSetSwingAngle function) in one of four modes while the setting gain (OBRSetGainControl function) is set up in Auto mode may cause a deterioration of scanning bar codes that are printed in a poor quality. If you need to set up the scan width control function, select the setting gain in Mode 0 to Mode 3. Or, if you select the setting gain in Auto mode, select the scan width control function in "No control on laser beam emission width" mode only.

4.93 OBRGetGainControl

This function retrieves gain control.

Calling Sequences

```
[C++]
int OBRGetGainControl(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetGainControl ( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetGainControl (
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is for retrieving the gain control. See **OBRSetGainControl** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.94 OBRSetBarWidthAdjustment

This function sets up correction value for adjusting the width of bars of scanned bar code.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetBarWidthAdjustment** function.

Calling Sequences

```
[C++]
int OBRSetBarWidthAdjustment (
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetBarWidthAdjustment ( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetBarWidthAdjustment (
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying correction value selecting one of the values listed below for adjusting the width of bars of scanned bar code. The default is "OBR_BAR_NORMAL".

OBR_BAR_NORMAL	: No correction.
OBR_BAR_BLACK0	: Thin black bars.
OBR_BAR_BLACK1	: Significantly thin black bars.
OBR_BAR_WHITE0	: Thin white bars.
OBR_BAR_WHITE1	: Significantly thin white bars.
OBR_BAR_BLACK2	: Thicken black bars.
OBR_BAR_BLACK3	: Significantly thicken black bars.
OBR_BAR_WHITE2	: Thicken white bars.
OBR_BAR_WHITE3	: Significantly thicken white bars.

Return Values

OBR_OK	: Normal end
OBR_PRN	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.95 OBRGetBarWidthAdjustment

This function retrieves the correction value for adjusting the width of bars of scanned bar code.

Calling Sequences

```
[C++]
int OBRGetBarWidthAdjustment (
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetBarWidthAdjustment ( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetBarWidthAdjustment (
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is pointer to DWORD-type variable that stores the setting of correction value for adjusting the width of bars of scanned bar code. See

OBRSetBarWidthAdjustment function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.96 OBRSetMarginCheckRatio

This function sets up threshold values for right and left side space margins for bar code symbol printed inside quadrangle.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetMarginCheckRatio** function.

Calling Sequences

```
[C++]
int OBRSetMarginCheckRatio(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetMarginCheckRatio ( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetMarginCheckRatio (
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying a threshold value for right and left side space margin selecting one of the values listed below.

OBR_MARGIN_WIDE	: Maximum (Default)
OBR_MARGIN_MIDDLE	: Middle
OBR_MARGIN_NARROW	: Narrow
OBR_MARGIN_MINIMUM	: Minimum



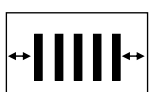
Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

Note:

When a bar code is printed inside of quadrangle, scanning may not be possible because there is not enough left and/or right marginal space. By making change on the right and left marginal threshold values, scanning a bar code becomes possible. See the table below.

Table 4.7

When the left marginal space is narrow.	
When the right marginal space is narrow.	
When both right and left marginal spaces are narrow.	

4.97 OBRGetMarginCheckRatio

This function retrieves the threshold value of the right and left side space margin for a bar code symbol printed inside quadrangle.

Calling Sequences

```
[C++]
int OBRGetMarginCheckRatio(
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetMarginCheckRatio ( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetMarginCheckRatio (
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is pointer to DWORD-type variable that stores the setting for the threshold value. See **OBRSetMarginCheckRatio** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.98 OBRSetDecodeLearningMode

This function sets up the scanner learning function.

In the Device Emulator, the function does not perform, but stores the preset value as internal variable. The value stored can be checked with **OBRGetDecodeLearningMode** function.

Calling Sequences

```
[C++]
int OBRSetDecodeLearningMode(
    DWORD dwMode
)
```

```
[Visual Basic]
Public Shared Function OBRSetDecodeLearningMode ( _
    ByVal dwMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRSetDecodeLearningMode (
    Int32 dwMode
);
```

Parameters

dwMode

This parameter is for specifying the scanner learning function selecting either of the values listed below. The default is "OBR_LEARNING_OFF".

OBR_LEARNING_OFF	: Scanner learning function to be set disabled
OBR_LEARNING_ON	: Scanner learning function to be set enabled

Return Values

OBR_OK	: Normal end
OBR_PRM	: Parameter error
OBR_PON	: Open already
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

4.99 OBRGetDecodeLearningMode

This function retrieves the setting for the Decoder Learning function.

Calling Sequences

```
[C++]
int OBRGetDecodeLearningMode (
    DWORD *pMode
)
```

```
[Visual Basic]
Public Shared Function OBRGetDecodeLearningMode ( _
    ByRef pMode As Int32 _
) As Int32
```

```
[C#]
public static Int32 OBRGetDecodeLearningMode (
    ref Int32 pMode
);
```

Parameters

pMode

This parameter is pointer to DWORD-type variable that stores the setting for the Decoder Learning function. See **OBRSetDecodeLearningMode** function for the values to retrieve.

Return Values

OBR_OK	: Normal end
OBR_NOT_DEVICE	: Scanner driver error. In the Device Emulator, this value is not returned.
FUNCTION_UNSUPPORTED	: Unsupported error

5. Notes to Programming

5.1 Scanning Completion Notification

A message or an event will be issued by the scanner driver integrated in the terminal when it has retrieved data (notification only by message has been set as default).

Message issued for the message notification uses WM_OBR_SCANNING(WM_USER +0x530), and one of the following values indicating a condition of scanning end will be saved in the *wParam* parameter of the message.

OBR_SUCCESS	: Normal end
OBR_TIMEOUT	: Timeout occurred
OBR_BUFFEROVER	: OBR buffer output error

If event notification mode is set enabled with **OBRSetScanningNotification** function, notification will be made with a named event. The named event is called "OBRScanningEvent". Also, by calling **OBRGetLastEventStatus** function after the event is issued, a factor that made the event to be issued can be retrieved. Factors of scanning end that are retrieved with **OBRGetLastEventStatus** function are those saved in the *wParam* parameter for the message notification.

Setting message notification

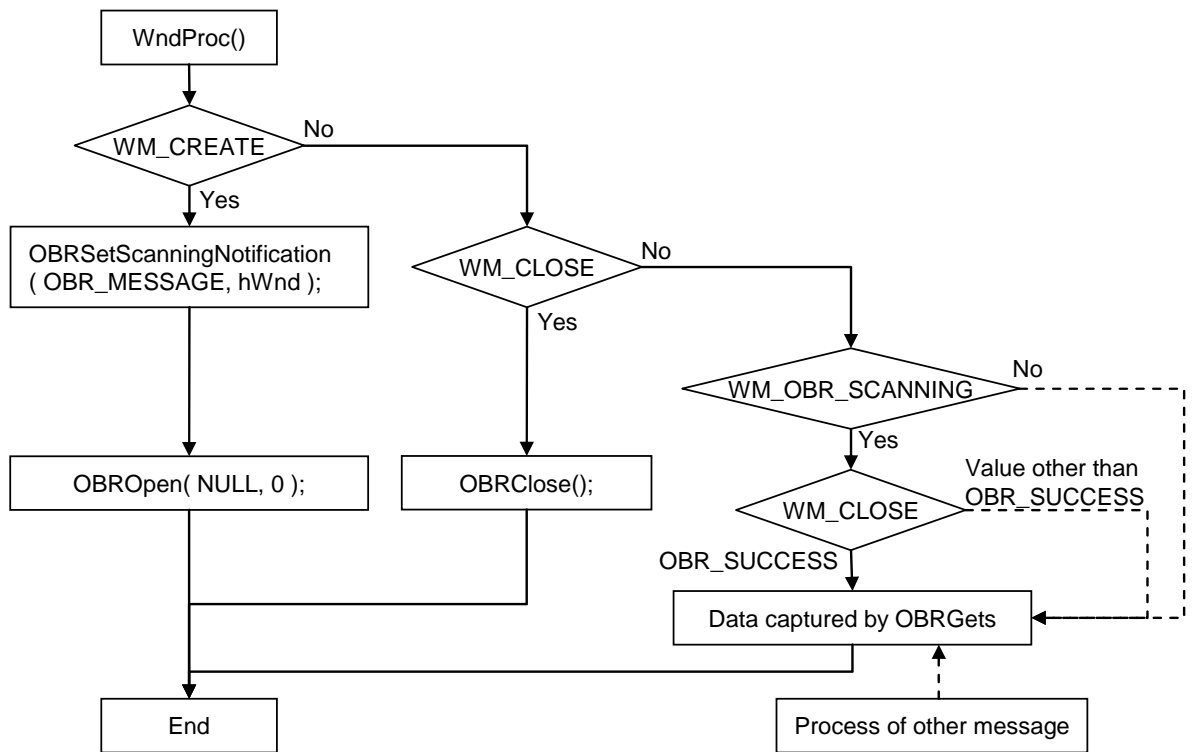


Figure 5.1 Flowchart

1. Notification mode for scanning completion is set to the message output with **OBRSetScanningNotification** function. At the same time, window handle of the destination which the message is sent to is specified.
2. After opening the scanner with **OBROpen** function, the flow waits until a scanning completion notification message is issued.
3. After receiving the message, the *wParam* parameter of the message is checked for its value. If scanning is succeeded, data output to each output destination will be processed.
4. Carry out **OBRClose** function to close the scanner.

Setting event notification

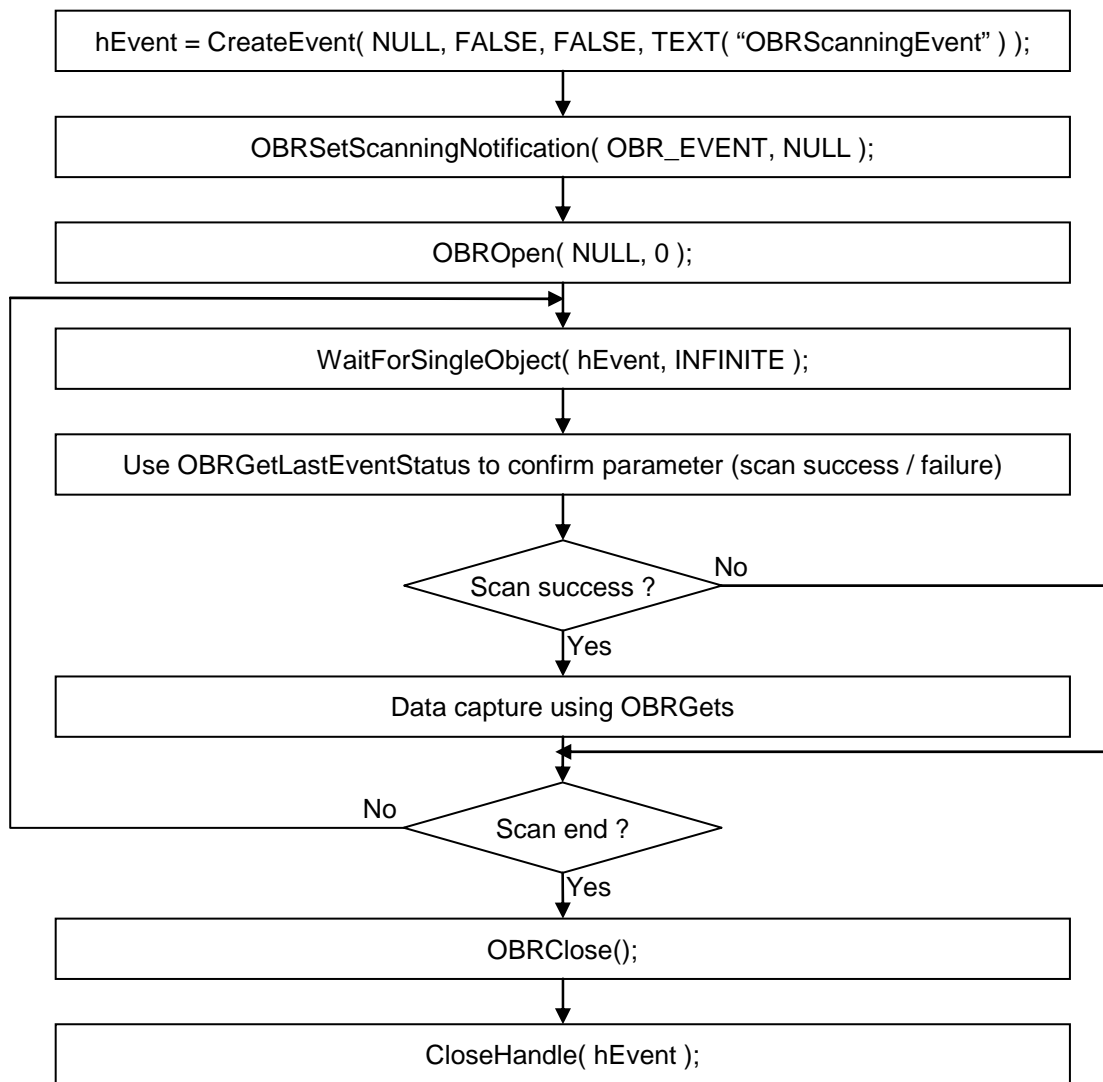


Figure 5.2 Flowchart

1. Named event is initialized.
2. Notification for scanning completion is set to event output with **OBRSetScanningNotification** function.
3. After opening the scanner with **OBROpen** function, the flow waits until named event is issued with `WaitForSingleObject`, etc.
4. After receiving the named event, conditions why the event is issued are checked with **OBRGetLastEventStatus** function.
5. If scanning is succeeded, data output to each output destination will be processed.
6. Carry out **OBRClose** function to close the scanner.

5.2 Configuration File Format

The setting parameters are saved in text format to the configuration file. The parameters saved are described in the table below. If any of the content in the file is changed using editor, it can be reflected to the scanner driver by calling **OBRLoadConfigFile** function and then

OBRSetDefaultSymbology function. For the precedence between parameter for symbology in the configuration file and parameter for the same symbology set up with the relevant library function, either one of the parameters that has been set right before running the parameter will have the precedence over the other parameter.

Table 5.1 Setting readable symbology parameters

Parameter	Description	Value/Meaning	
READCODE	"Enable" or "Disable" for scanning with each symbology. For enabling multiple symbologies, specify a sum of the values in logical OR for each symbology. (Note 1)	1	Enable Code39.
		2	Enable NW-7.
		4	Enable WPC Addon.
		8	Enable WPC.
		16	Enable UPC-E Addon.
		32	Enable UPC-E.
		64	Enable IDF.
		128	Enable ITF.
		256	Enable Code93.
		512	Enable Code128.
		1024	Enable MSI.
		2048	Enable IATA.
		4096	Enable RSS-14
		8192	Enable RSS Limited
		16384	Enable RSS Expanded
		32768	Enable RSS-14 Stacked
		65536	Enable RSS Expanded Stacked

Table 5.2 Code39 option parameters

CODE39_MIN	Minimum no. of digits for scanning.	1 to 47 (DT-X11) 1 to 52 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
CODE39_MAX	Maximum no. of digits.	1 to 47 (DT-X11) 1 to 52 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
CODE39_OUTFORM AT	Specify output format.	0	With start and stop codes
		1	Without start and stop codes
		2	With Full ASCII conversion and start and stop codes
		3	With Full ASCII conversion, but without start and stop codes
CODE39_CHKD	Specify check digit calculation	0	Disable
		1	Enable

CODE39_CHKCH	Specify check character output	0	Disable
		1	Enable

Table 5.3 NW-7 option parameters

NW7_MIN	Specify minimum no. of digits.	1 to 62 (DT-X11) 1 to 63 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
NW7_MAX	Specify maximum no. of digits.	1 to 62 (DT-X11) 1 to 63 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
NW7_OUTFORMAT	Specify output format.	0	With start and stop codes
		1	Without start and stop codes
NW7_CHKD	Specify check digit calculation.	0	Disable
		1	Enable(DT-X7, DT-X30, IT-800, IT-300, DT-X8)
NW7_CHKCH	Specify check character output.	0	Disable
		1	Enable

Table 5.4 WPC Addon option parameters

WPCADDON_MIN	Specify minimum no. of digits.	10 (Fixed)	
WPCADDON_MAX	Specify maximum no. of digits.	18 (Fixed)	
WPCADDON_OUTFORMAT	Specify output format.	0	Output WPC Addon symbology with "0" at the forefront.
		1	Output WPC-Addon symbology without "0" at the forefront.
WPCADDON_CHKD (Note 5)	Specify check digit calculation.	0	Disable
		1	Enable
WPCADDON_CHKCH	Specify check character output.	0	Disable (Fixed)

Table 5.5 WPC option parameters

WPC_MIN	Specify minimum no. of digits.	8 (Fixed)	
WPC_MAX	Specify maximum no. of digits.	13 (Fixed)	
WPC_OUTFORMAT	Specify output format.	0	Output WPC with "0" at the forefront.
		1	Output WPC without "0" at the forefront.
		14	Output WPC in GTIN format.
WPC_CHKD (Note 5)	Specify check digit calculation.	0	Disable
		1	Enable
WPC_CHKCH	Specify check character output.	0	Disable (Fixed)

Table 5.6 UPCE Addon option parameters

UPCEADDON_MIN	Specify minimum no. of digits.	9 (Fixed)	
UPCEADDON_MAX	Specify maximum no. of digits.	12 (Fixed)	
UPCEADDON_OUTFORM AT	Specify output format.	0	"0" at the forefront is output.
		1	"0" at the forefront is not output.
UPCEADDON_CHKD (Note 5)	Specify check digit calculation.	0	Disable
		1	Enable
UPCEADDON_CHKCH	Specify check character output.	0	Disable
		1	Enable

Table 5.7 UPCE option parameters

UPCE_MIN	Specify minimum no. of digits.	7 (Fixed)	
UPCE_MAX	Specify maximum no. of digits.	7 (Fixed)	
UPCE_OUTFORMAT	Specify output format.	0	Output UPC-E with "0" at the forefront.
		1	Output UPC-E without "0" at the forefront.
		2	Output UPC-E in GTIN format.
UPCE_CHKD (Note 5)	Specify check digit calculation.	0	Disable
		1	Enable
UPCE_CHKCH	Specify check character output.	0	Disable
		1	Enable

Table 5.8 IDF option parameters

IDF_MIN	Specify minimum no. of digits.	2 to 55 (DT-X11) 2 to 67 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
IDF_MAX	Specify maximum no. of digits.	2 to 55 (DT-X11) 2 to 67 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
IDF_OUTFORMAT	Specify output format.	0	No output format (Fixed)
IDF_CHKD	Specify check digit calculation.	0	Disable
		1	Enable
IDF_CHKCH	Specify check character output.	0	Disable
		1	Enable

Table 5.9 ITF option parameters

ITF_MIN	Specify minimum no. of digits.	2 to 94 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7) 2 to 92 (DT-X11)	
ITF_MAX	Specify maximum no. of digits.	2 to 94 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7) 2 to 92 (DT-X11)	
ITF_OUTFORMAT	Specify output format.	0	No output format (Fixed)
ITF_CHKD	Specify check digit calculation.	0	Disable
		1	Enable
ITF_CHKCH	Specify check character output.	0	Disable
		1	Enable

Table 5.10 CODE93 option parameters

CODE93_MIN	Specify minimum no. of digits.	1 to 77 (DT-X11) 1 to 70 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
CODE93_MAX	Specify maximum no. of digits.	1 to 77 (DT-X11) 1 to 70 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
CODE93_OUTFORMAT	Specify output format.	0	No output format (Fixed)
CODE93_CHKD	Specify check digit calculation.	0	Disable
		1	Enable
CODE93_CHKCH	Specify check character output.	0	Disable (Fixed)

Table 5.11 CODE128 option parameters

CODE128_MIN	Specify minimum no. of digits.	1 to 98 (DT-X11, IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8)	
CODE128_MAX	Specify maximum no. of digits.	1 to 98 (DT-X11, IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8)	
CODE128_OUTFORMAT (Note 3)	Specify output format.	0	With Full ASCII conversion
		1	Without Full ASCII conversion
		2	EAN-128 only.
		6	EAN-128 only with ID code.
		10	Convert Fnc to GS and output EAN-128 only.
	Specify output format. (Note 6)	2	EAN-128 only.
		4	EAN-128 with ID code.
		8	Convert Fnc to GS and output EAN-128 only.
		16	Code128 without Full ASCII conversion.
		32	EAN-128 without Full ASCII conversion.
		64	Code128 and EAN-128 FNC2 combined.
		128	Code128 and EAN-128 FNC4 expansion ASCII conversion.
CODE128_CHKD (Note 5)	Specify check digit calculation.	0	Disable
		1	Enable
CODE128_CHKCH	Specify check character output.	0	Disable (Fixed)

Table 5.12 MSI option parameters

MSI_MIN	Specify minimum no. of digits.	1 to 67 (DT-X11) 1 to 57 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
MSI_MAX	Specify maximum no. of digits.	1 to 67 (DT-X11) 1 to 57 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
MSI_OUTFORMAT	Specify output format.	0	Without output format
MSI_CHKD (Note 4)	Specify check digit calculation.	0	Without check digit calculation
		1	1 digit : mod 10
		2	2 digits 1st: mod 11, 2nd: mod 10
		3	2 digits 1st: mod 10, 2nd: mod 10
MSI_CHKCH	Specify check character output.	0	Disable
		1	Enable

Table 5.13 IATA option parameters

IATA_MIN	Specify minimum no. of digits.	1 to 44 (DT-X11) 1 to 65 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
IATA_MAX	Specify maximum no. of digits.	1 to 44 (DT-X11) 1 to 65 (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)	
IATA_OUTFORMAT	Specify output format.	0	Disable (Fixed)
IATA_CHKD	Specify check digit calculation.	0	Disable
		1	Calculate all digits except the last digit.
		2	Calculate coupon no. and data.
		3	Calculate data section only.
		4	Modulus10
IATA_CHKCH	Specify check character output.	0	Disable (Fixed)

Table 5.14 RSS-14 option parameters

RSS14_MIN	Specify minimum no. of digits.	14 (Fixed)	
RSS14_MAX	Specify maximum no. of digits.	14 (Fixed)	
RSS14_OUTFORMAT	Specify output format.	0	Output RSS-14 symbology in standard format.
		1	Output RSS-14 symbology without "01" at the forefront.
RSS14_CHKD	Specify check digit calculation.	1	Enable (Fixed)
RSS14_CHKCH	Specify check character output.	1	Enable (Fixed)

Table 5.15 RSS Limited option parameters

RSSLTD_MIN	Specify minimum no. of digits.	14 (Fixed)	
RSSLTD_MAX	Specify maximum no. of digits.	14 (Fixed)	
RSSLTD_OUTFORMAT	Specify output format.	0	Output RSS Limited symbology in standard format.
		1	Output RSS Limited symbology without "01" at the forefront.
RSSLTD_CHKD	Specify check digit calculation.	1	Enable (Fixed)
RSSLTD_CHKCH	Specify check character output.	1	Enable (Fixed)

Table 5.16 RSS Expanded option parameters

RSSEXP_MIN	Specify minimum no. of digits.	1 to 74	
RSSEXP_MAX	Specify maximum no. of digits.	1 to 74	
RSSEXP_OUTFORMAT	Specify output format.	0	No output (Fixed)
RSSEXP_CHKD	Specify check digit calculation.	1	Enable (Fixed)
RSSEXP_CHKCH	Specify check character output.	1	Enable (Fixed)

Table 5.17 RSS-14 Stacked option parameters (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8)

RSS14S_MIN	Specify minimum no. of digits.	14 (Fixed)	
RSS14S_MAX	Specify maximum no. of digits.	14 (Fixed)	
RSS14S_OUTFORMAT	Specify output format.	0	Output RSS-14 Stacked symbology in standard format.
		1	Output RSS-14 Stacked symbology without "01" at the forefront.
RSS14S_CHKD	Specify check digit calculation.	1	Enable (Fixed)
RSS14S_CHKCH	Specify check character output.	1	Enable (Fixed)

Table 5.18 RSS Expanded Stacked option parameters (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8)

RSSEXPS_MIN	Specify minimum no. of digits.	1 to 74	
RSSEXPS_MAX	Specify maximum no. of digits.	1 to 74	
RSSEXPS_OUTFORMAT	Specify output format.	0	No output (Fixed)
RSSEXPS_CHKD	Specify check digit calculation.	1	Enable (Fixed)
RSSEXPS_CHKCH	Specify check character output.	1	Enable (Fixed)

Table 5.19 Read mode option parameters

READMODE	Specify scanning mode.	0	Single scan
		1	Continuous scan

Table 5.20 Gain option parameters (IT-600, DT-X7, DT-X30 IT-800, IT-300, DT-X8) (Note 7)

GAIN	Specify a gain mode.	0	Mode 0
		1	Mode 1
		2	Mode 2
		3	Mode 3
		128	Gain Auto (Note 8) (DT-X7, IT-800, IT-300, and DT-X8)

Table 5.21 Buzzer control option parameters

BUZZER	Set buzzer setting.	0	Disable
		1	Enable

Table 5.22 LED control option parameters

LEDCTRL	Set LED setting.	0	Disable
		1	Enable
		2	Enable Disable when scan fails

Table 5.23 OutBuff control option parameters

OUTBUFF	Specify output mode.	0	Output to OBR buffer.
		2	Output to Key.
		3	Output to Clipboard.
		4	Output to Keyboard event.

Table 5.24 Suff Char option parameters

ENDCODE	Specify termination code.	0	CR
		1	LF
		2	CR+LF
		3	TAB
		4	No termination code

Table 5.25 Multi-step read option parameters

MULTISTEP	Specify multi-step read mode.	0	Normal read
		1	Multi-step read

Table 5.26 Comp Counter option parameters

CMPCNT	Specify no. of verifications.	In the range of 1 to 9(Laser) In the range of 0 to 9(Linear Imager)	
--------	-------------------------------	--	--

Table 5.27 Read counter option parameters

READCNT	Specify no. of times to scan in continuous read mode.	In the range of 1 to 9	
---------	---	------------------------	--

Table 5.28 Scan time option parameters

SCANTIME	Specify a time period (in second) of timeout.	In the range of 1 to 9	
FILTERMODE	Specify noise filter mode.	0	Disable noise filter.
		1	Enable software filter.
		2	Enable hardware filter
		3	Enable software/hardware filters.
FILTERCNT	Specify a time span between when decoding starts and until when noise filter begins to activate.	In the range of 1 to 8	

Table 5.29 Vibrator control option parameters (IT-600 and DT-X7)

VIBRATOR	Set vibration.	0	Disable
		1	Enable

Table 5.30 Scanning key option parameters

KEY	Specify scanning trigger keys. For specifying multiple keys, specify a sum of the values in logical OR for specified keys.	1	L Trigger key
		2	R Trigger key
		4	Multi key
		8	L Cursor key
		16	R Cursor key
		32	Up Cursor key
		64	Down Cursor key
		128	Trigger Grip key (IT-600)
		256	Center Trigger key

Table 5.31 Laser beam swing angle option parameters (IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)

SWING	Specify laser beam swing angle.	0	No control
		1	Wide
		2	Medium
		3	Narrow
		(Note 8)	

Table 5.32 Laser focus option parameters (IT-600, DT-X7, DT-X30, IT-800) (Note 7)

FOCUS	Set laser focus.	0	Disable
		1	Enable

Table 5.33 Notification option parameters

NOTIFICATION	Specify notification mode for scanning completion.	1	Window message notification
		2	Event notification

Table 5.34 Decode level option parameters (DT-X11, IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8)

LEVEL	Set decode level.	0	Standard (normal)
		1	High

Table 5.35 Decode Customize parameters (DT-X11, IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)

BARWIDTH	Specify a correctness of bar's thickness or thinness.	0	No correction.
		1	Correct black bar thin.
		2	Correct black bar thin noticeably.
		3	Correct white bar thin.
		4	Correct white bar thin noticeably.
		5	Correct black bar thick.
		6	Correct black bar thick noticeably.
		7	Correct white bar thick.
		8	Correct white bar thick noticeably.
MARGINCHECK	Specify a threshold for right and left margins.	0	Large
		1	Medium
		2	Small
		3	Minimum

Table 5.36 Learning Decode option parameters (DT-X11, IT-600, DT-X7, DT-X30, IT-800, IT-300, DT-X8) (Note 7)

LEARNING	Specify "Enable" or "Disable" for the learning decode function.	0	Disable
		1	Enable

Notes:

- To set multiple symbologies effect for reading, specify a sum of the values in logical OR for each symbology you wish to scan.
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support the following.
 - Reading UPC-B and UPC-B Addon symbologies.
 - Setting concurrently for reading both Industrial 2of5 and IATA symbologies. If it is done mistakenly, setting for reading Industrial 2of5 becomes enabled, but, setting for the other symbology becomes disabled.
 - Individual settings for reading RSS-14, RSS Expanded, RSS-14 Stacked, and RSS Expanded Stacked symbologies. If the standard RSS symbology is set enabled, all the corresponded Stacked symbologies are automatically set enabled.
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support the outputs of Code128 and EAN-128 symbologies without ASCII conversion, and the FNC2 joint output to be disabled.
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)" and "OBR_CDMSEV (With 2-digit check digit calculation)" to be set in the respective parameters.
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support "OBR_CHKDOFF (Without check digit calculation)" to be set in **byCheckDigit** parameter.
- To concurrently specify multiple output conditions effect, specify a sum of the values for each condition you wish to set effect. However, the concurrent setting of "EAN-128 without Full ASCII conversion" with any one of "EAN-128 with ID code", "Convert Fnc to GS and output EAN-128", "Code128 and EAN-128 FNC2 combined", or "Code128 and EAN-128 FNC4 expansion ASCII conversion" effect is not supported. If it is done wrongly, the setting on "EAN-128 without Full ASCII conversion" is preferentially set enabled.
- IT-600M30U, IT-600M30UC, DT-X7M10U, and IT-300E-35E do not support the parameter.
- Setting up the scan width control function (OBRSetSwingAngle function) in one of four modes while the setting gain (OBRSetGainControl function) is set up in Auto mode may cause a

deterioration of scanning bar codes that are printed in a poor quality. If you need to set up the scan width control function, select the setting gain in Mode 0 to Mode 3. Or, if you select the setting gain in Auto mode, select the scan width control function in "No control on laser beam emission width" mode only.