

# TM-S9000/S2000

## API Reference Guide for .NET

---

### Overview

Descriptions of the features and a development environment.

### Creating a Development Environment

Descriptions of construction methods for development environments

### Programming Guide

Descriptions of a programming method for application development using the TM-S9000/S2000 API.

### TM-S9000/S2000 API Reference

Describes the TM-S9000/S2000 API.

## Cautions

- No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Seiko Epson Corporation.
- The contents of this document are subject to change without notice. Please contact us for the latest information.
- While every precaution has taken in the preparation of this document, Seiko Epson Corporation assumes no responsibility for errors or omissions.
- Neither is any liability assumed for damages resulting from the use of the information contained herein.
- Neither Seiko Epson Corporation nor its affiliates shall be liable to the purchaser of this product or third parties for damages, losses, costs, or expenses incurred by the purchaser or third parties as a result of: accident, misuse, or abuse of this product or unauthorized modifications, repairs, or alterations to this product, or (excluding the U.S.) failure to strictly comply with Seiko Epson Corporation's operating and maintenance instructions.
- Seiko Epson Corporation shall not be liable against any damages or problems arising from the use of any options or any consumable products other than those designated as Original EPSON Products or EPSON Approved Products by Seiko Epson Corporation.

## Copyright notice

EPSON is a registered trademark of Seiko Epson Corporation.

Exceed Your Vision and ESC/POS are registered trademarks or trademarks of Seiko Epson Corporation.

MS-DOS<sup>®</sup>, Microsoft<sup>®</sup>, Win32<sup>®</sup>, Windows<sup>®</sup>, Windows Vista<sup>®</sup>, Windows Server<sup>®</sup>, Visual Studio<sup>®</sup>, Visual Basic<sup>®</sup>, Visual C++<sup>®</sup>, and Visual C#<sup>®</sup> are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

EPSON TM-S9000/S2000 Driver is based in part on the work of the Independent JPEG Group.

libtiff

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Intel<sup>®</sup> Core<sup>™</sup> and Pentium<sup>®</sup> are registered trademarks of Intel Corporation.

All other trademarks are the property of their respective owners and used for identification purpose only.

Copyright © 2012-2014 Seiko Epson Corporation. All rights reserved.

# For Safety

## Key to Symbols

The symbols in this manual are identified by their level of importance, as defined below. Read the following carefully before handling the product.

	Provides information that must be observed to avoid damage to your equipment or a malfunction.
	Provides important information and useful tips.

## Restriction of Use

When this product is used for applications requiring high reliability/safety such as transportation devices related to aviation, rail, marine, automotive etc.; disaster prevention devices; various safety devices etc; or functional/precision devices etc, you should use this product only after giving consideration to including fail-safes and redundancies into your design to maintain safety and total system reliability. Because this product was not intended for use in applications requiring extremely high reliability/safety such as aerospace equipment, main communication equipment, nuclear power control equipment, or medical equipment related to direct medical care etc, please make your own judgment on this product's suitability after a full evaluation.

# About this Manual

## Aim of the Manual

This manual is aimed at development engineers and provides necessary information for developing a .NET application using TM-S9000/S2000 driver.

Information for this product is provided in the TM-S9000/S2000 API Reference Guide for Win32/64.

## Manual Content

The manual is made up of the following sections:

- Chapter 1      [Overview](#)
- Chapter 2      [Creating a Development Environment](#)
- Chapter 3      [Programing Guide](#)
- Chapter 4      [TM-S9000/S2000 API Reference](#)

# Contents

■ <b>For Safety</b> .....	<b>3</b>
Key to Symbols .....	3
■ <b>Restriction of Use</b> .....	<b>3</b>
■ <b>About this Manual</b> .....	<b>4</b>
Aim of the Manual .....	4
Manual Content .....	4
■ <b>Contents</b> .....	<b>5</b>

---

## Overview ..... 15

■ <b>Manual Overview</b> .....	<b>15</b>
Reference Information .....	15
■ <b>Operating Environment</b> .....	<b>16</b>
OS .....	16
.NET Framework .....	16
Computer .....	17
Interface .....	17
■ <b>Development Environment</b> .....	<b>18</b>
■ <b>Technical Support Web Site</b> .....	<b>18</b>

---

## Creating a Development Environment ..... 19

■ <b>Configuring references</b> .....	<b>19</b>
■ <b>Declaration Statement</b> .....	<b>20</b>
When using Visual C# .....	20
When using Visual Basic .NET .....	20

---

## Programing Guide ..... 21

■ <b>Reference Information</b> .....	<b>21</b>
■ <b>How to use API</b> .....	<b>21</b>
Method and Property .....	21
Event Callback Registration Method .....	22
■ <b>Programming flow</b> .....	<b>24</b>
Check scan processing .....	24
Card scan processing .....	26
Electronic endorsement printing .....	28
Physical endorsement printing .....	30
Roll paper printing (TM-S9000 Only) .....	33
Template printing .....	35
Cut sheet paper printing .....	37

■ Priority for error detection.....	40
■ Actions to be taken in the event of errors.....	42
Device status.....	42
Return value.....	43
■ Sample Program.....	46

---

## **TM-S9000/S2000 API Reference..... 47**

■ API Comparison Chart (Alphabetical Order).....	47
■ Class list.....	55
■ API Index.....	56
Basic APIs.....	56
API List (Alphabetical Order).....	57
■ MFDevice Class -Constructor- .....	59
■ ResetLastError.....	60
■ OpenMonPrinter .....	61
■ CloseMonPrinter .....	63
■ SCNMICRCancelStatusBack .....	63
■ StartEndorsementCancelStatusBack .....	64
■ EndEndorsementCancelStatusBack.....	64
■ GetRealStatus .....	65
■ CancelStatusBack.....	66
■ CancellnkStatusBack .....	66
■ MICRSelectDataHandling.....	67
■ SCNSelectScanUnit .....	69
■ SCNSetImageTypeOption .....	70
■ SCNGetImageTypeOption .....	73
■ SCNMICRFunctionContinuously.....	74
■ SCNMICRFunctionPostPrint .....	79
■ SCNMICRFunction .....	80
■ SCNMICRCancelFunction .....	84
■ SCNDeleteCroppingArea.....	85
■ SCNSelectScanFace .....	86
■ GetPrnCapability .....	87
■ SCNGetImageFormat .....	88
■ SCNSetImageFormat .....	89
■ SCNGetScanArea .....	90
■ SCNSetScanArea.....	91
■ SCNGetImageQuality.....	92
■ SCNSetImageQuality .....	93

■ SCNSelectScanImage .....	95
■ SCNGetCroppingAreas .....	96
■ SCNSetCroppingArea .....	97
■ SCNGetImageAdjustment.....	98
■ SCNSetImageAdjustment.....	100
■ MICRClearSpaces.....	102
■ SetOcrABAreaOrigin.....	103
■ GetMicrText.....	104
■ GetOcrABText .....	105
■ GetScanImage.....	106
■ GetBarcodeData.....	107
■ DecodeBarcode .....	108
■ DecodeBarcodeMemory .....	109
■ SetMonInterval .....	110
■ GetPrintStation.....	111
■ SetPrintStation.....	112
■ GetPrintPosition .....	113
■ SetPrintPosition.....	114
■ SetEndorseDirection.....	115
■ BufferedPrint.....	116
■ GetPrintSize.....	117
■ GetPrintControl.....	118
■ SetPrintControl .....	119
■ SetPrintSize .....	120
■ GetPrintImageMethod.....	121
■ SetPrintImageMethod.....	122
■ PrintBarCode.....	123
■ PrintImage .....	127
■ GetPrintAlignment.....	128
■ SetPrintAlignment.....	129
■ PrintMemoryImage.....	130
■ SCNPrintMemoryImage.....	131
■ PrintText .....	132
■ SCNPrintText .....	133
■ PrintMultipleToneImage.....	134
■ UpdateEndorseText .....	135
■ InsertValidation.....	136
■ RemoveValidation.....	137

■ GetPrintCutSheetSettings.....	138
■ SetPrintCutSheetSettings.....	139
■ PrintCutSheet .....	140
■ AutoCutRollPaper.....	143
■ LoadTemplatePrintArea .....	144
■ TemplatePrint .....	145
■ SetTemplatePrintArea .....	147
■ ClearTemplatePrintData .....	149
■ GetTransactionNumber .....	150
■ SetTransactionNumber .....	151
■ SetTransactionNumberWithIncremental .....	152
■ SetWaterfallMode.....	153
■ GetIQAResult .....	154
■ GetVersion .....	155
■ ESCNEnable .....	156
■ ESCNGetAutoSize.....	157
■ ESCNSetAutoSize .....	158
■ ESCNGetCutSize .....	159
■ ESCNSetCutSize .....	160
■ ESCNGetRotate.....	161
■ ESCNSetRotate.....	162
■ ESCNGetDeSkew .....	163
■ ESCNSetDeSkew .....	164
■ ESCNGetDocumentSize .....	165
■ ESCNSetDocumentSize .....	166
■ ESCNDefineCropArea .....	167
■ ESCNGetMaxCropAreas.....	168
■ ESCNStoreImage .....	169
■ ESCNClearImage .....	170
■ ESCNRetrieveImage.....	171
■ ESCNGetRemainingImages .....	172
■ SetBehaviorToScnResult .....	173
■ SetNumberOfDocuments .....	174
■ SelectErrorEjectAtContinuously.....	175
■ SelectJamDetect.....	176
■ MICRGetStatus.....	177
■ MICRCleaning .....	178
■ InkHeadCleaning.....	179

■ <b>OpenDrawer</b> .....	180
■ <b>RingBuzzer</b> .....	181
■ <b>GetCounter</b> .....	182
■ <b>ResetCounter</b> .....	183
■ <b>LoadAPISettings</b> .....	184
■ <b>SetConfigure</b> .....	185
■ <b>ResetPrinter</b> .....	186
■ <b>CancelError</b> .....	187
■ <b>GetType</b> .....	188
■ <b>GetOfflineCodeByIndex</b> .....	189
■ <b>SCNGetClampStatus</b> .....	190
■ <b>SetPaperThickness</b> .....	191
■ <b>Properties</b> .....	192
LastError .....	192
IsValid.....	192
Status .....	192
InkStatus .....	193
OfflineCode .....	193
ESCNAutoSize .....	193
ESCNCutSize .....	193
ESCNRotate .....	194
ESCNDocumentSize .....	194
ESCNDeSkew .....	194
ESCNRemainingImages.....	194
TransactionNumber .....	195
TransactionIncrement.....	195
PrintStation.....	195
PrintSize .....	196
PrintPosition.....	196
PrintAlignment.....	196
■ <b>Events</b> .....	197
StatusCallback/StatusCallbackEx.....	197
InkStatusCallback/InkStatusCallbackEx .....	198
MfDone .....	198
SCNMICRStatusCallback.....	199
StartEndorsementStatusCallback .....	199
EndEndorsementStatusCallback.....	200
■ <b>MFBase Class - Properties</b> .....	201
Version .....	201
Ret.....	201
Timeout.....	201
ErrorEject .....	202
SuccessEject .....	202
UseNVMemory .....	202
BuzzerHz(3) .....	203
BuzzerCount(3) .....	203
PortName.....	204

<b>■ MFMicr Class - Properties .....</b>	<b>205</b>
Version.....	205
Ret .....	205
Font .....	205
MicrOcrSelect .....	205
Parsing .....	206
Status.....	206
Detail.....	207
MicrStr .....	208
AccountNumber .....	208
Amount .....	208
BankNumber .....	209
SerialNumber.....	209
EPC .....	209
TransitNumber .....	209
CheckType .....	209
OcrReliableInfo.FirstSelectString .....	210
OcrReliableInfo.SecondSelectString .....	210
OcrReliableInfo.FirstSelectPercentage .....	210
OcrReliableInfo.SecondSelectPercentage .....	210
OnUSField .....	210
AuxillaryOnUSField .....	211
<b>■ MFScan Class - Properties .....</b>	<b>212</b>
Version.....	212
Ret .....	212
ImageID .....	212
Resolution .....	213
AddInfoData .....	213
Image.....	213
Data .....	213
Status.....	214
Detail.....	214
<b>■ MFDeviceFont Class - Constructor .....</b>	<b>215</b>
<b>■ MFDeviceFont Class - Properties .....</b>	<b>216</b>
Font .....	216
Size .....	216
Underline .....	216
Bold .....	217
Color .....	217
Reverse .....	217
<b>■ MFTrueType Class - Constructor .....</b>	<b>218</b>
<b>■ MFTrueType Class - Properties .....</b>	<b>219</b>
Font .....	219
Color .....	219
Reverse .....	219
<b>■ MFProcess Class - Properties .....</b>	<b>220</b>
Version.....	220
ActivationMode.....	220
PaperType .....	221
StartWaitTime .....	221
SuccessStamp.....	221
PaperMisInsertionErrorSelect .....	222
PaperMisInsertionErrorEject .....	222

PaperMisinsertionStamp.....	222
PaperMisinsertionCancel.....	223
NoiseErrorSelect.....	223
NoiseErrorEject.....	223
NoiseStamp.....	224
NoiseCancel.....	224
DoubleFeedErrorSelect.....	224
DoubleFeedErrorEject.....	225
DoubleFeedStamp.....	225
DoubleFeedCancel.....	225
BaddataErrorSelect.....	226
BaddataCount.....	226
BaddataErrorEject.....	226
BaddataStamp.....	227
BaddataCancel.....	227
NodataErrorSelect.....	227
NodataErrorEject.....	228
NodataStamp.....	228
NodataCancel.....	229
NearFullSelect.....	229
ResultPartialData.....	229
EndorsePrintMode.....	230
PrnDataLenExceedErrorEject.....	230
PrnDataLenExceedCancel.....	231
PrnDataUnreceiveErrorEject.....	231
PrnDataUnreceiveCancel.....	231
<b>■ MFIq Class - Properties.....</b>	<b>232</b>
Version.....	232
ErrorSelect.....	232
ErrorEject.....	232
Stamp.....	233
Cancel.....	233
ImageFormat.....	234
ColorDepth.....	234
Threshold.....	234
Color.....	235
ExOption.....	235
Resolution.....	236
Undersize.....	236
Oversize.....	237
Mincompressed.....	237
Maxcompressed.....	237
Front_rear.....	238
Toolight.....	238
Toodark.....	238
Streaks.....	239
Noise.....	239
Focus.....	239
Corners.....	240
Edges.....	240
Framing.....	240
Skew.....	241
Carbon.....	241
Piggyback.....	241

<b>■ MFImageResult Class - Properties</b> .....	<b>242</b>
Version.....	242
Ret .....	242
UndersizeImage .....	243
OversizeImage .....	243
MaxCompressedImageSize .....	244
MinCompressedImageSize.....	244
FrontRearImageMismatch .....	245
ImageTooLight .....	245
ImageTooDark .....	246
HorizontalStreaksPresent.....	246
ExcessiveSpotNoise .....	247
ImageOutOfFocus.....	247
FoldedTornDocCorners.....	248
FoldedTornDocEdges.....	249
DocFramingError .....	250
ExcessiveDocSkew.....	250
CarbonStripDetection.....	251
Piggyback .....	251
<b>■ MFBarcode Class - Properties</b> .....	<b>252</b>
Version.....	252
Ret .....	252
ErrorSelect.....	252
ErrorEject.....	253
Stamp.....	253
Cancel.....	254
TargetColor.....	254
Resolution .....	255
InfoMode .....	255
Info.....	256
Data .....	256
<b>■ MFBarcode.BarcodeInfo Class - Properties</b> .....	<b>257</b>
Ret .....	257
Status.....	257
Detail.....	258
SymbolMask .....	258
Direction .....	259
Origin.....	259
StartX.....	260
StartY.....	260
EndX .....	260
EndY .....	260
<b>■ MFBarcode.BarcodeData Class - Properties</b> .....	<b>261</b>
Symbol .....	261
Direction .....	261
StartX.....	262
StartY.....	262
EndX .....	262
EndY .....	262
Data .....	262

■ <b>MFOcrAb Class - Properties</b> .....	<b>263</b>
Version .....	263
Ret.....	263
OcrType.....	264
Direction.....	264
StartX .....	265
StartY .....	265
EndX.....	265
EndY.....	266
SpaceHandling .....	266
OcrStr.....	266
OcrReliableInfo.FirstSelectString .....	267
OcrReliableInfo.SecondSelectString.....	267
OcrReliableInfo.FirstSelectPercentage.....	267
OcrReliableInfo.SecondSelectPercentage .....	267
■ <b>MFVersion Class - Properties</b> .....	<b>268</b>
Description.....	268
Version .....	268
■ <b>MFPrintAreaInfo Class - Properties</b> .....	<b>269</b>
AreaName .....	269
OriginX.....	269
OriginY.....	269
Width .....	269
Height .....	270
Rotate.....	270
Measure .....	270
■ <b>MFPrint Class - Properties</b> .....	<b>271</b>
Version .....	271
Ret.....	271
EndorsementType .....	271
String(3) .....	272
Attribute(3).....	272
Font(3) .....	272
Speed .....	273
Direction.....	273
■ <b>Device Information</b> .....	<b>274</b>
Device Status.....	274
Ink Status .....	277
Maintenance Counter .....	278
MICR Status.....	279
Offline Code .....	280
Device ID .....	292
Type ID.....	294



# Overview

This chapter describes the features and specifications of the product.

## Manual Overview

This manual provides information regarding the development of .NET applications. For more details about this product, refer to the TM-S9000/S2000 API Reference Guide for Win32/64.

### Reference Information

Refer to the TM-S9000/S2000 API Reference Guide for Win32/64 for the following information.

Content	Reference	
	Chapter	Item
TM-S9000/S2000 API features	1	<ul style="list-style-type: none"> <li>• Features</li> <li>• Features of the TM-S9000/S2000 API</li> </ul>
How to install and uninstall a driver.	2	Install and Uninstall
TM-S9000/S2000 API Reference (TM-S9000/S2000 API details )	4	TM-S9000/S2000 API Reference
About the information on compatibility of the TM-J9000 API/TM-S1000 API.	5	Compatible Information
About the configuration file for the TM-S9000/S2000 driver.	6	Setting File
How to generate a log file.	7	Log Function

# Operating Environment

## OS

- Microsoft Windows 8.1 (32/64 bit)
- Microsoft Windows 8 (32/64 bit)
- Microsoft Windows 7 SP1 (32/64 bit)
- Microsoft Windows Vista SP2 (32/64 bit)
- Microsoft Windows XP SP3 (32 bit)
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2012
- Microsoft Windows Server 2008 SP2 (32/64 bit)
- Microsoft Windows Server 2008 R2 SP1
- Microsoft Windows Embedded Standard 7 (32/64bit)
- Microsoft Windows XP Embedded (32bit)
- Microsoft Windows Embedded POS Ready 7 (32/64bit)
- Microsoft Windows Embedded POS Ready 2009 (32bit)

## .NET Framework

- Microsoft .NET Framework 2.0 SP2
- Microsoft .NET Framework 3.0 SP2
- Microsoft .NET Framework 3.5 SP1
- Microsoft .NET Framework 4
- Microsoft .NET Framework 4.5

## Computer

---

### ***110 dpm model***

- CPU : At least a Pentium 4 2.0 GHz or the equivalent  
Memory : At least 512 MB or above the minimum operating system requirement  
HDD : At least 30 MB Free space. (Before installing the driver)

---

### ***200 dpm model***

- CPU : At least Intel Core 2 Duo 1.8GHz or the equivalent  
Memory : At least 1 GB or above the minimum operating system requirement  
HDD : At least 30 MB Free space. (Before installing the driver)

## Interface

USB 2.0

# Development Environment

Development Tool	Development Language
Visual Studio .NET 2003	Visual Basic .NET 2003
	Visual C# .NET 2003
Visual Studio 2005	Visual Basic 2005
	Visual C# 2005
Visual Studio 2008	Visual Basic 2008
	Visual C# 2008
Visual Studio 2010	Visual Basic 2010
	Visual C# 2010

## Technical Support Web Site

For customers in North America, go to the following web site:

<http://www.epsonexpert.com/>

For customers in other countries, go to the following web site:

<https://download.epson-biz.com/?service=pos>

# Creating a Development Environment

This chapter describes how to create the environment such as Visual Studio for developing the application.

## Configuring references

Before developing the application, configure references in Visual Studio.



When moving the TM-J9000/TM-S1000 application, its reference configuration must be replaced with DLL provided for the TM-S9000/S2000 API. The following explains the reference configuration procedure using Visual C# of Visual Studio 2010. The same procedure is required when using other development environments.

Configure references using the following procedures:

- 1 Launch a new project.
- 2 From the (Project) menu, select (Add Reference).



When moving the TM-J9000/TM-S1000 application, delete the current DLL settings before adding new references.

- 3 The Add Reference screen is displayed. Click on the Reference tab and select "EpsonBankDriver.dll", then click (OK). The "EpsonBankDriver.dll" has been stored in /Sample programs/Assembly.
- 4 Open the Solution Explorer to be sure that the component file "EpsonBankDriver.dll" added at step 3 has been properly added in the project. This is the end of the reference configuration procedure for .NET environment.

# Declaration Statement

The component file configured as reference must be declared to be used for actual programming. The declaration statement differs between Visual C# and Visual Basic .NET.

## When using Visual C#

The following declaration should be added in declaration section of the source code.

Declaration:

```
using com.epson.bank.driver
```

## When using Visual Basic .NET

The following declaration should be added in declaration section of the source code.

Declaration:

```
Imports com.epson.bank.driver
```

# Programming Guide

This chapter describes a programming method for the development of application using TM-S9000/S2000 .NET API.

## Reference Information

For the following information as it applies to .NET, refer to Chapter 3 of the TM-S9000/S2000 API Reference Guide for Win32/64.

- Application Processing Steps
- Actions to be taken in the event of errors

## How to use API

### Method and Property

The TM-S9000/S2000 .NET API consists of Method and Property. Some of the functions provided in the Method and Property are similar to each other, however, the way to use the functions are different. For more details on the API, refer to [TM-S9000/S2000 API Reference \(p.47\)](#).

#### **How to use Method**

Specify parameters for the Method to be used. Details of the parameters specified and execution results for each method, refer to [TM-S9000/S2000 API Reference \(p.47\)](#).

```
ErrorCode = Method (Parameter1, Parameter2, ...);
```

#### **How to use Property**

Use the Property as described below.

- 1 Acquire or set a value using the Property.  
Information on the values, refer to [TM-S9000/S2000 API Reference \(p.47\)](#).

##### **Acquire a value:**

```
Value = <CLASS>.Property;
```

##### **Set a value:**

```
<CLASS>.Property = Value;
```

- 2 API execution results are set in the LastError property.  
See [LastError \(p.192\)](#) for information on the API execution results.

```
ErrorCode = <MFDevice CLASS>.LastError;
```

## Event Callback Registration Method

Under .NET environment, events are provided instead of the CALLBACK functions provided under Win32/64 environment.

For more details on the events, refer to [Events \(p.197\)](#). For more details on the CALLBACK functions, refer to TM-S9000/S2000 API Reference Guide for Win32/64 - Chapter 3 "Programming guide".

---

### Example (When the scan processing)

Definition

```
ErrorCode SCNMICRSetStatusBack()
```

Delegate Definition

```
delegate void SCNMICRStatusCallbackHandler  
(int transactionNumber,  
  MainStatus mainStatus,  
  ErrorCode subStatus,  
  String portName )
```

Event Definition

```
event SCNMICRStatusCallbackHandler SCNMICRStatusCallback
```

Using

#### Application callback function definition

```
public void AppSCNMICRStatusHandler(int transactionNumber,  
  MainStatus mainStatus,  
  ErrorCode subStatus,  
  String portName )  
  
{  
}
```

#### Registration procedure

- 1** Registration procedure.  
This does not enable an event to be issued.

```
<MFDevice>.SCNMICRStatusCallback +=  
  new MFDevice.SCNMICRStatusCallbackHandler  
  (AppSCNMICRStatusHandler);
```

- 2** Function that enables an event to be issued.  
Calling this function allows an event to be issued.

```
<MFDevice>.SCNMICRSetStatusBack();
```

**Cancel procedure**

- 1** Disables an event to be issued.

```
<MFDevice>.SCNMICRCancelStatusBack();
```

- 2** Cancels the registered callback function.

```
<MFDevice>.SCNMICRStatusCallback -=  
    new MFDevice.SCNMICRStatusCallbackHandler  
    (AppSCNMICRStatusHandler);
```

# Programming flow

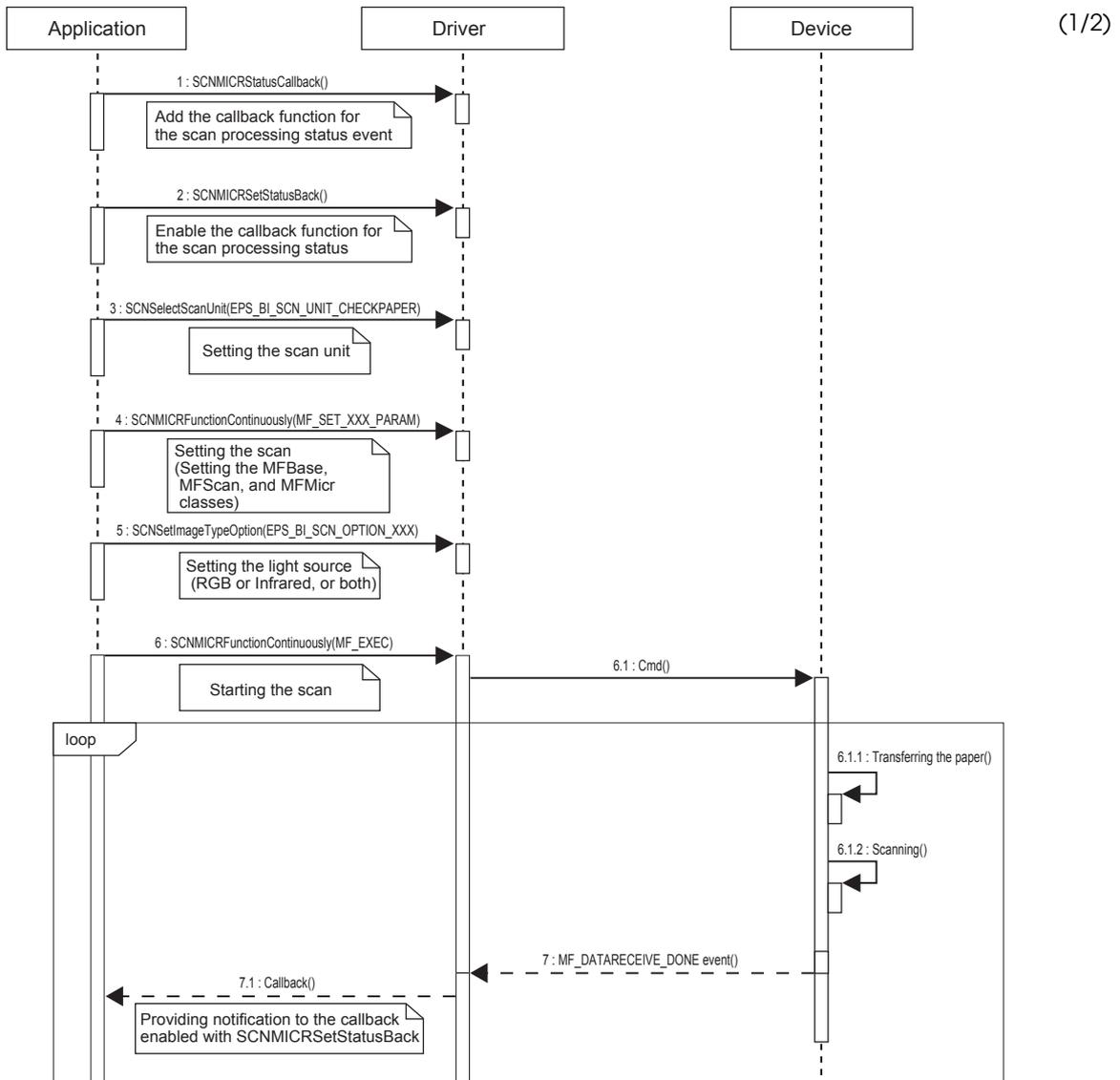
This section describes how to program the features offered by the TM-S9000/S2000 .NET API by using sequence diagrams.

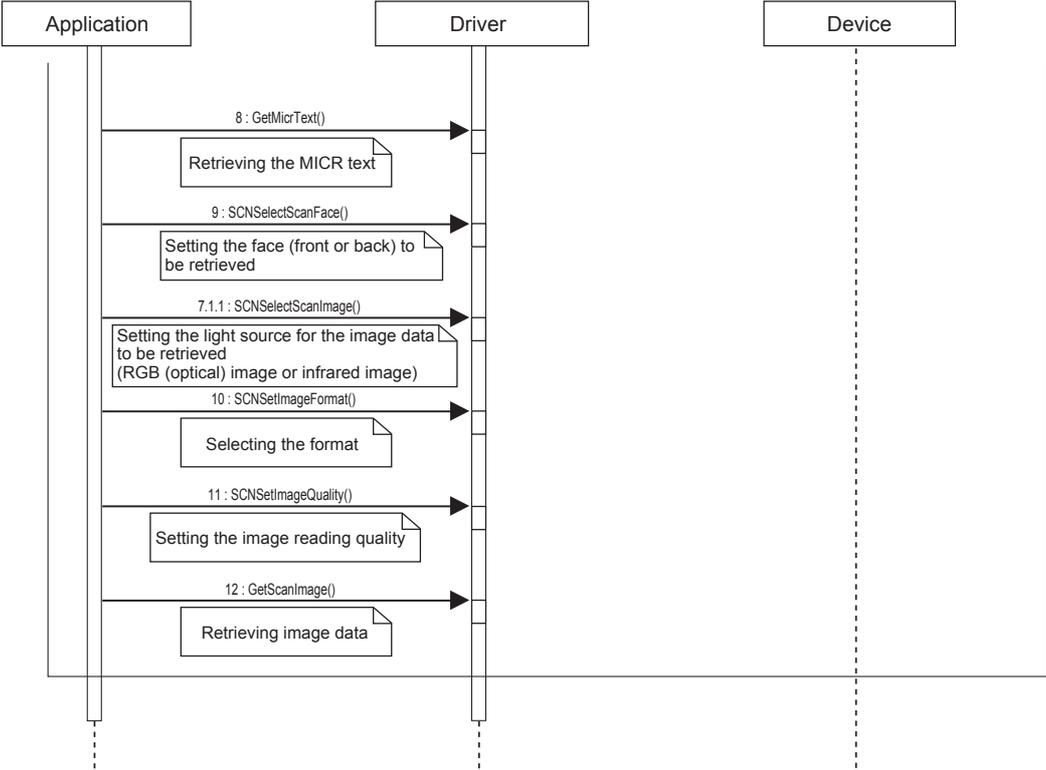


Information on DLL loading, constructor of MF\_Device class, OpenMonPrinter and CloseMonPrinter are not provided here.

## Check scan processing

This is the process for scanning the check sheet and retrieving the scan results (MICR data, image data).

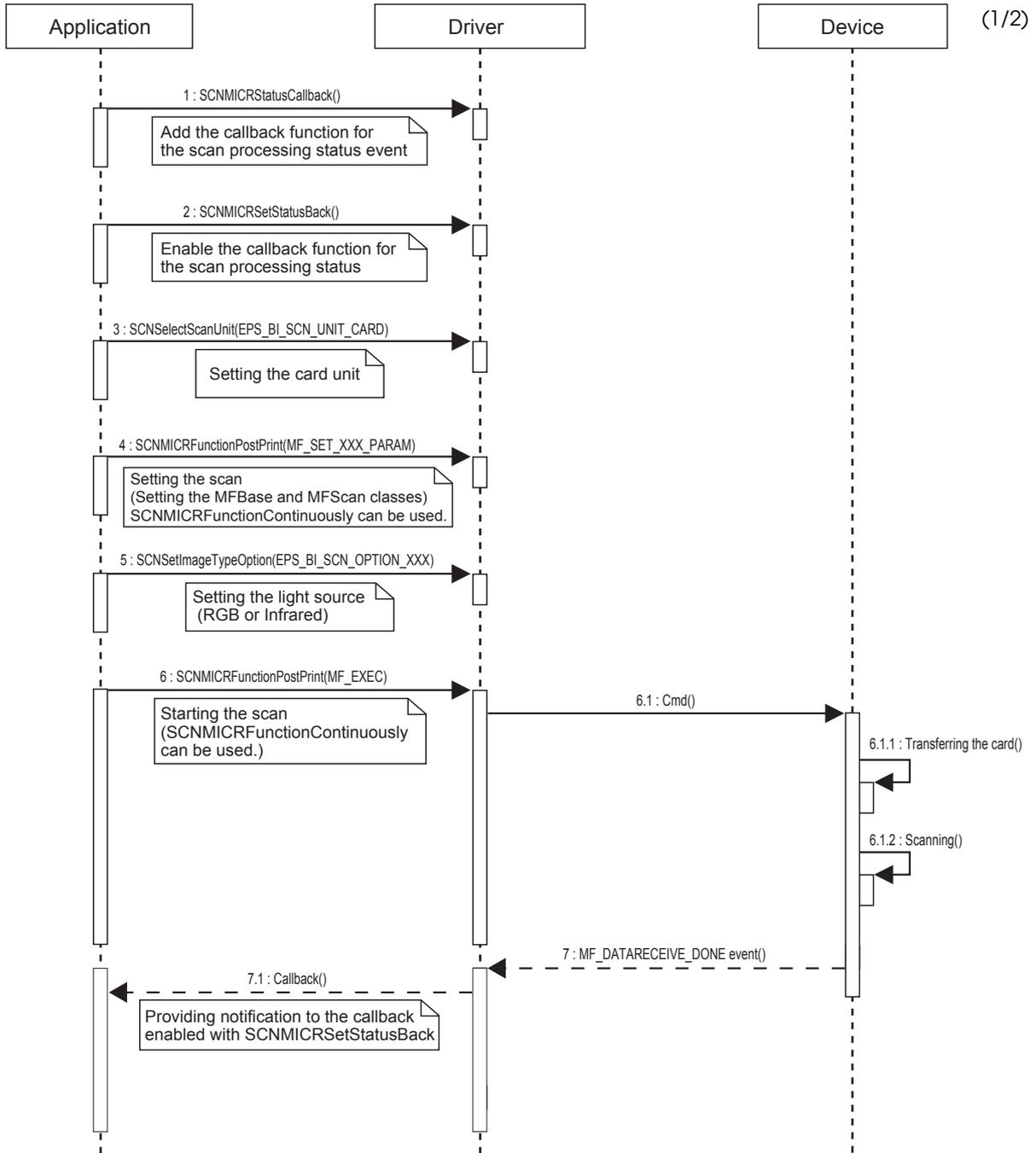


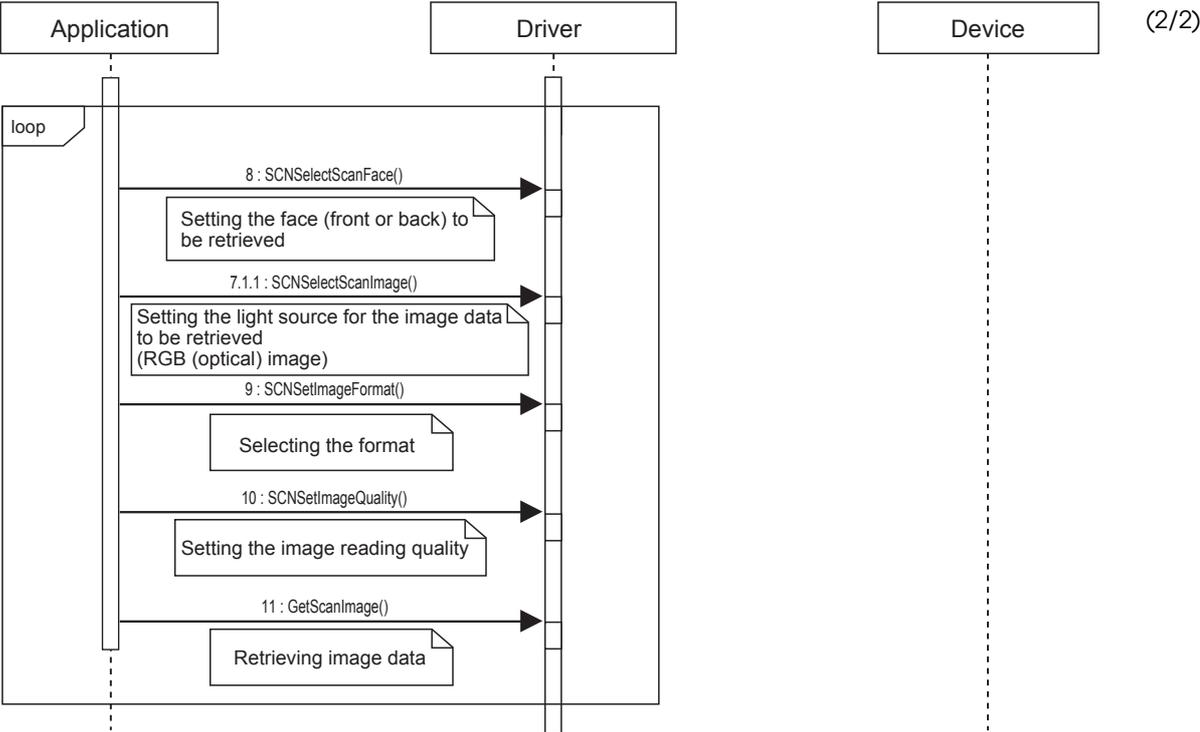


(2/2)

## Card scan processing

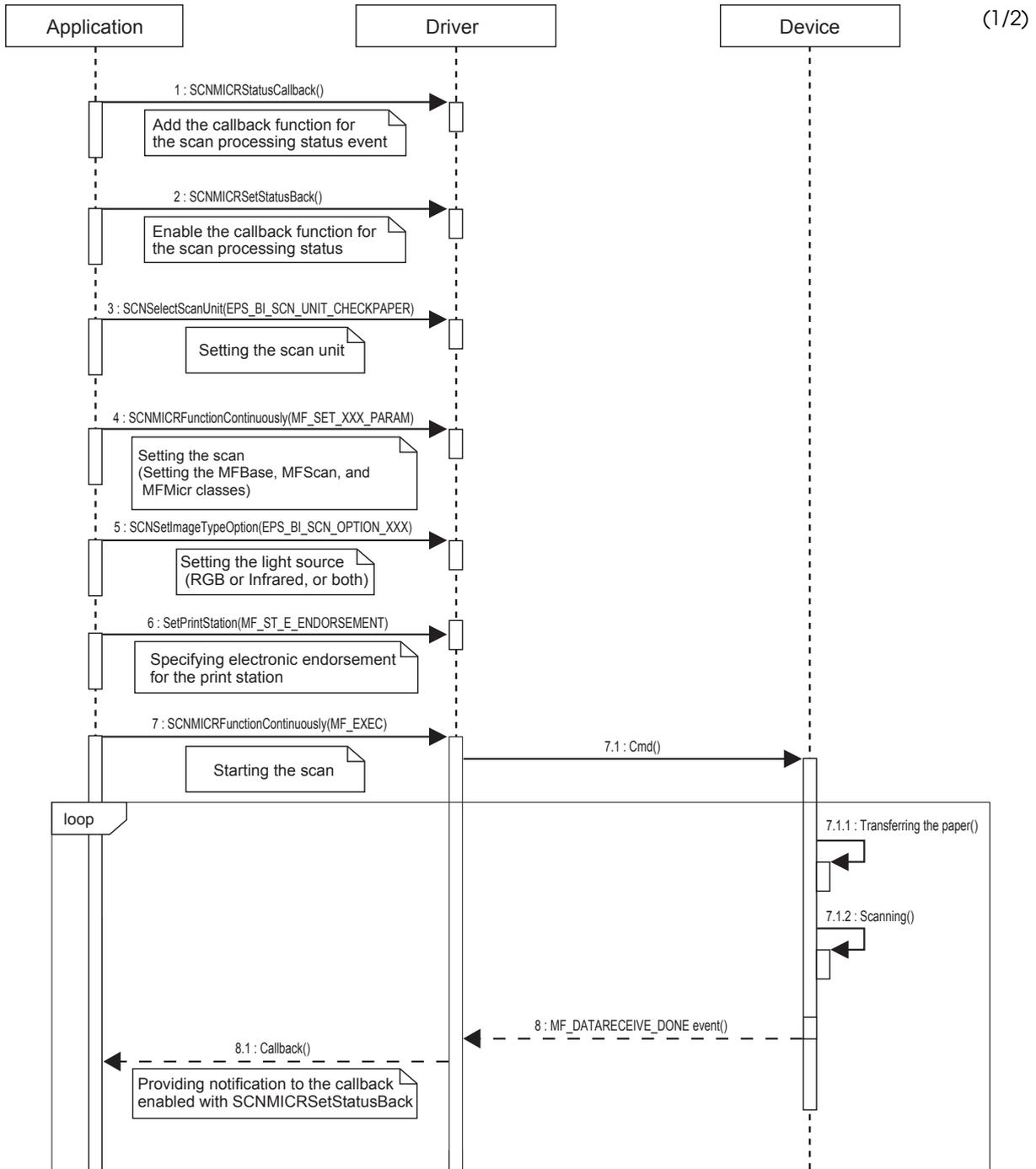
This is the process for scanning a card and retrieving the scan results (image data).

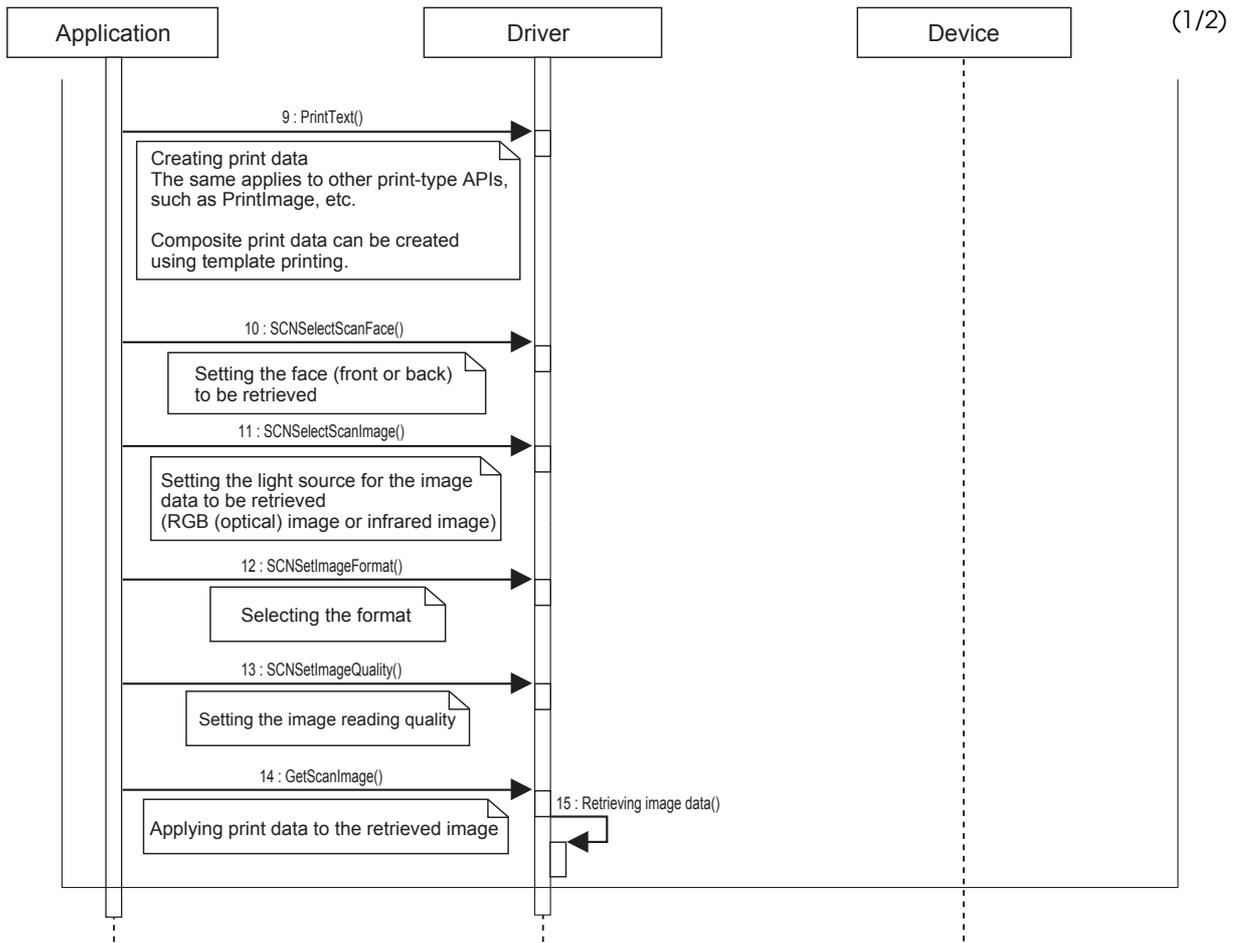




## Electronic endorsement printing

This is the process for scanning the check sheet and adding an electronic endorsement to retrieve the scan results (image data).

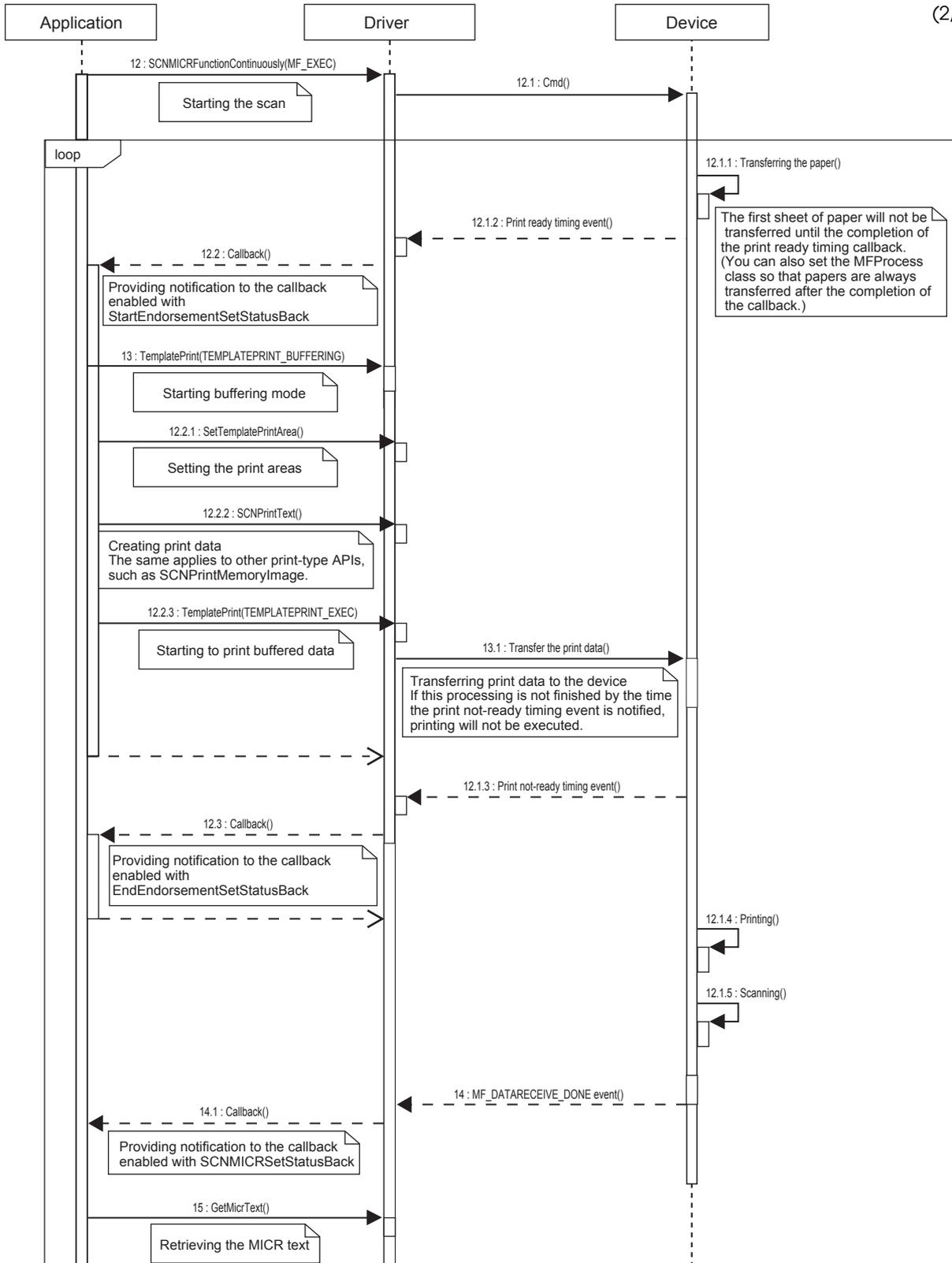


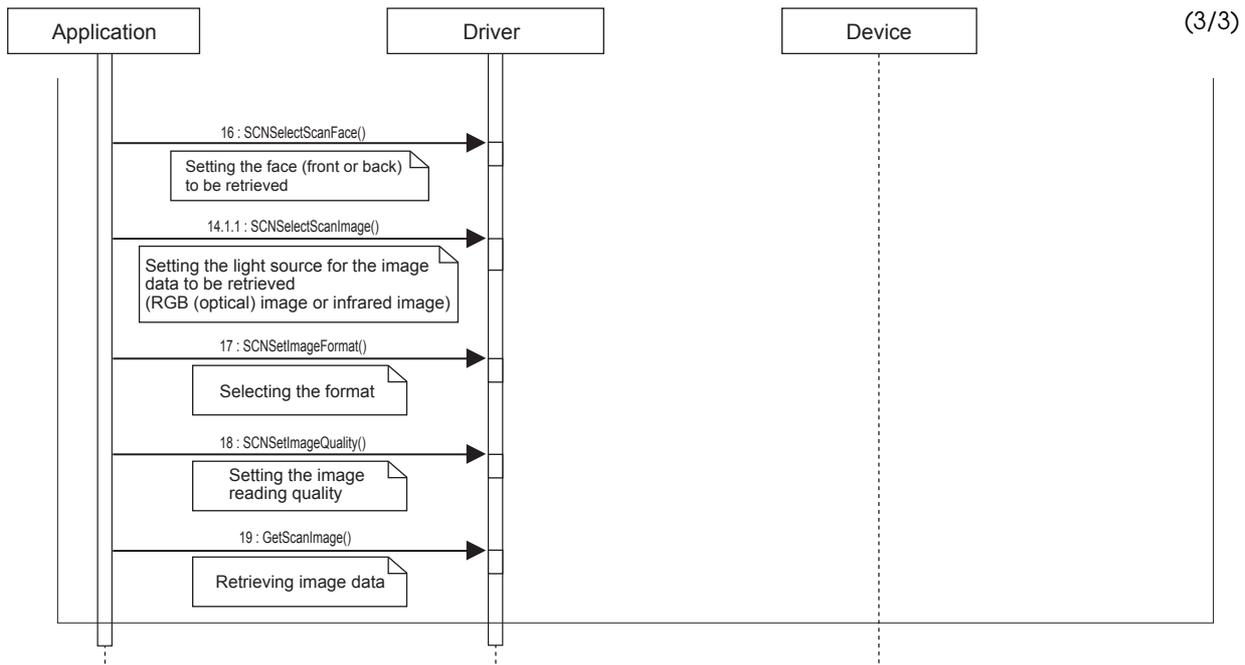


## Physical endorsement printing

This is the process for applying a physical endorsement and then scanning the check sheet, and then retrieving the scan results (image data).

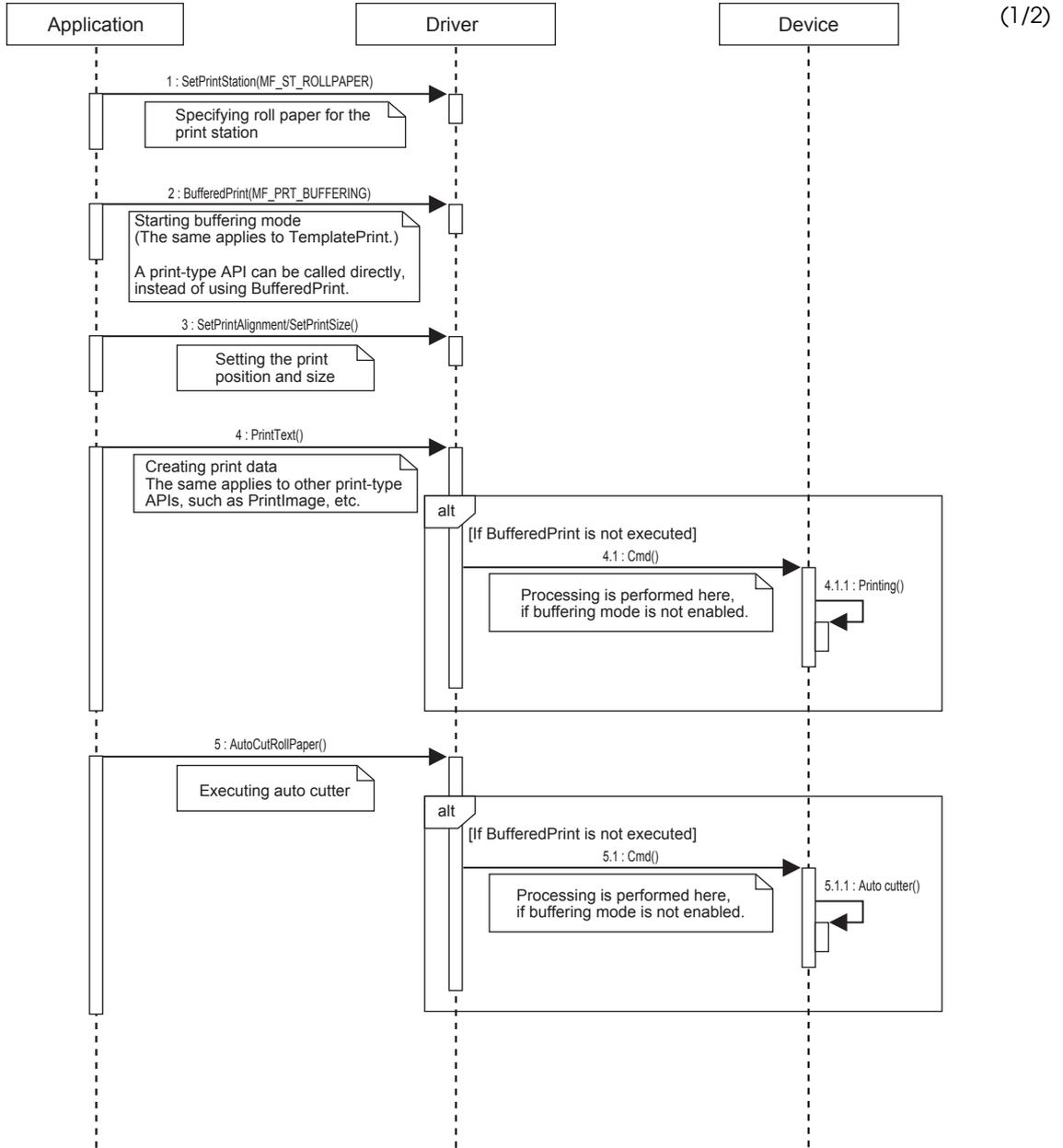


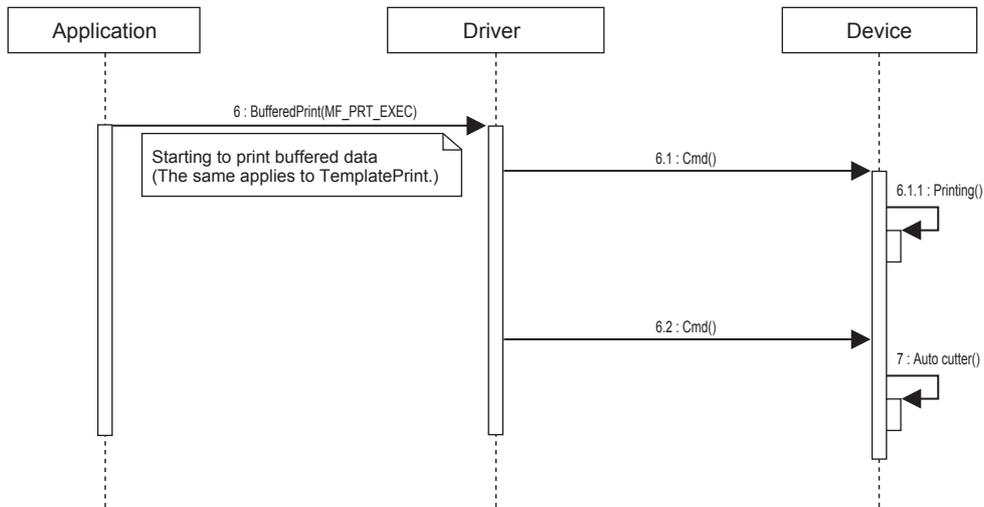




## Roll paper printing (TM-S9000 Only)

This is the process for printing roll paper.



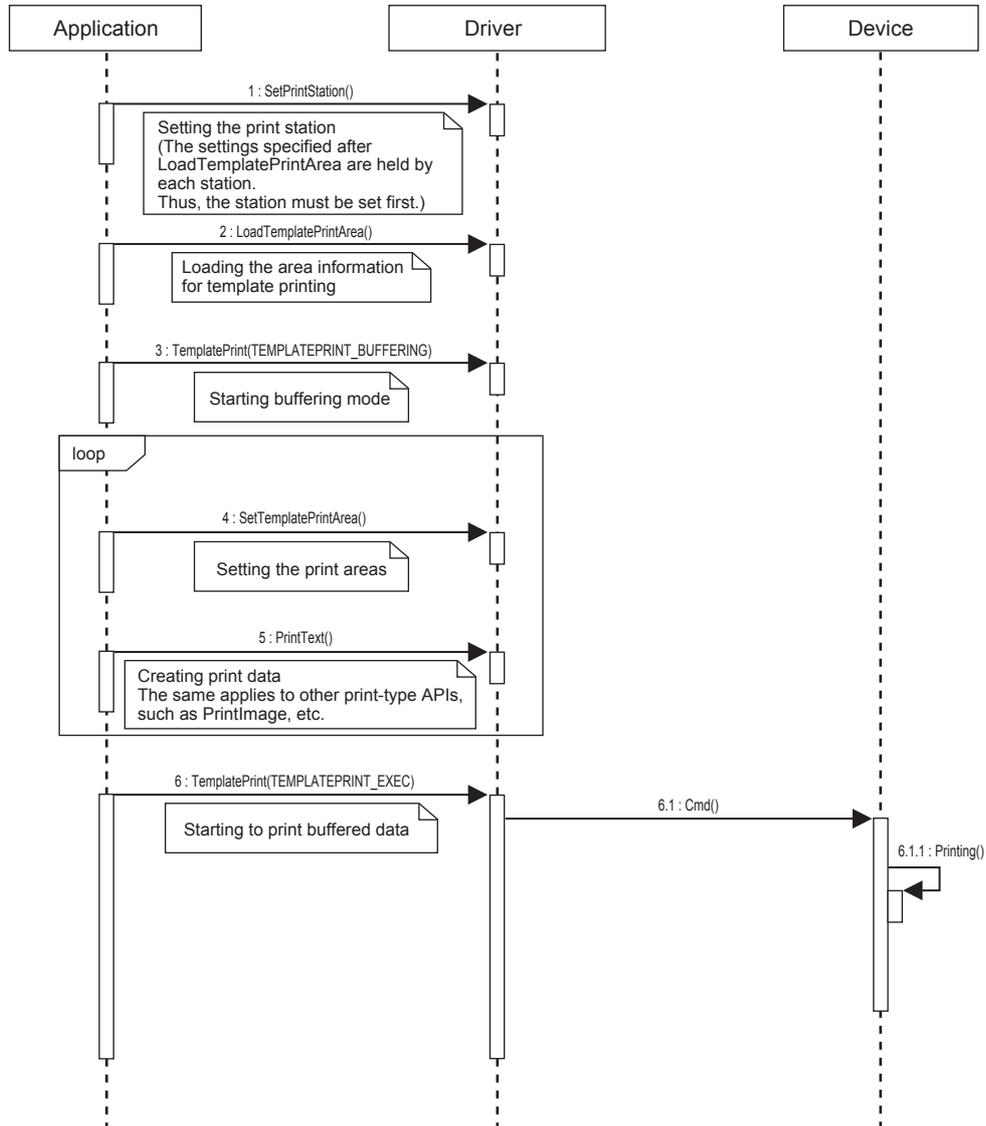


## Template printing

This is the process for specifying an area for printing.

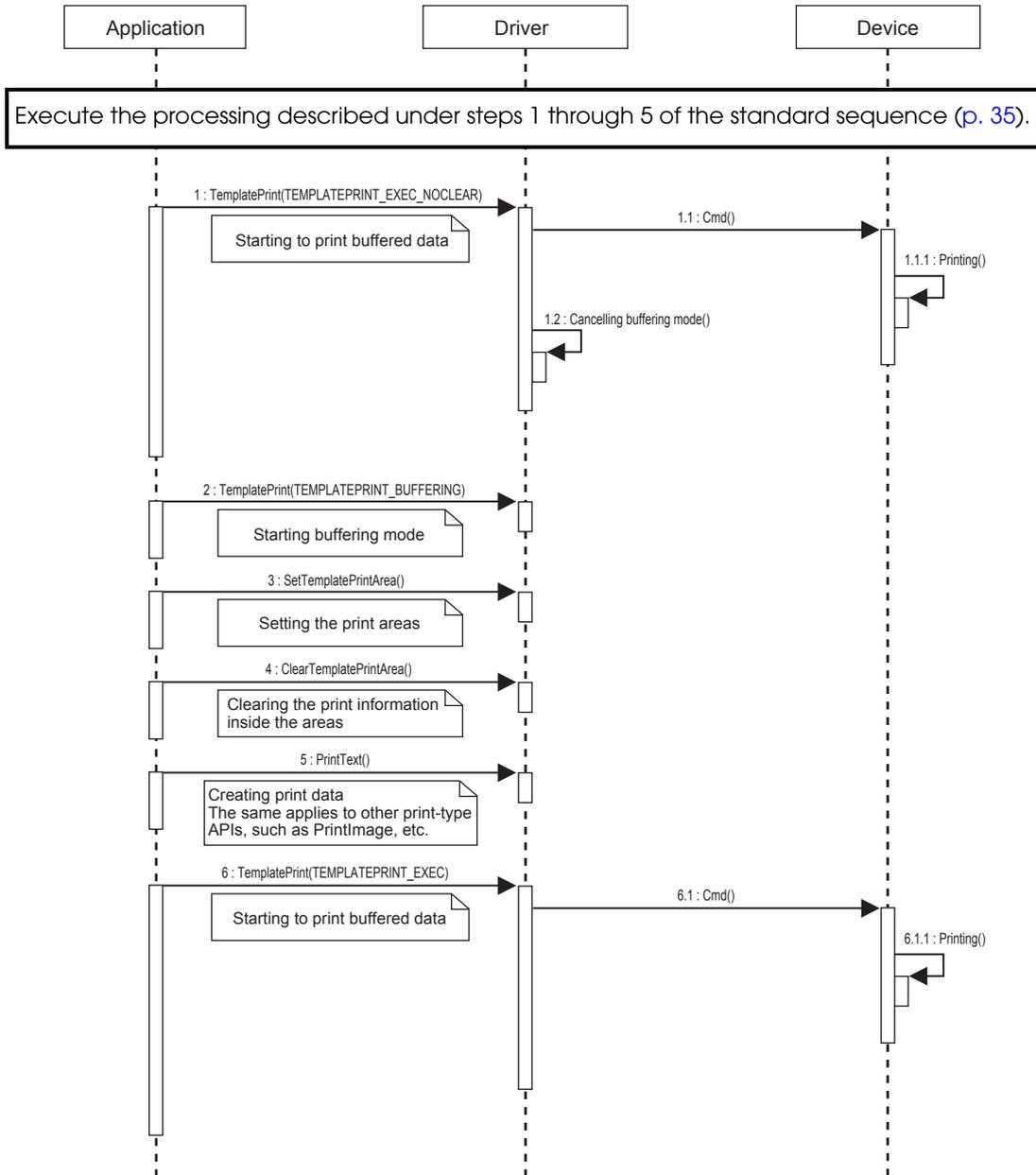
### Standard sequence

This sequence is a basic sequence for template printing.



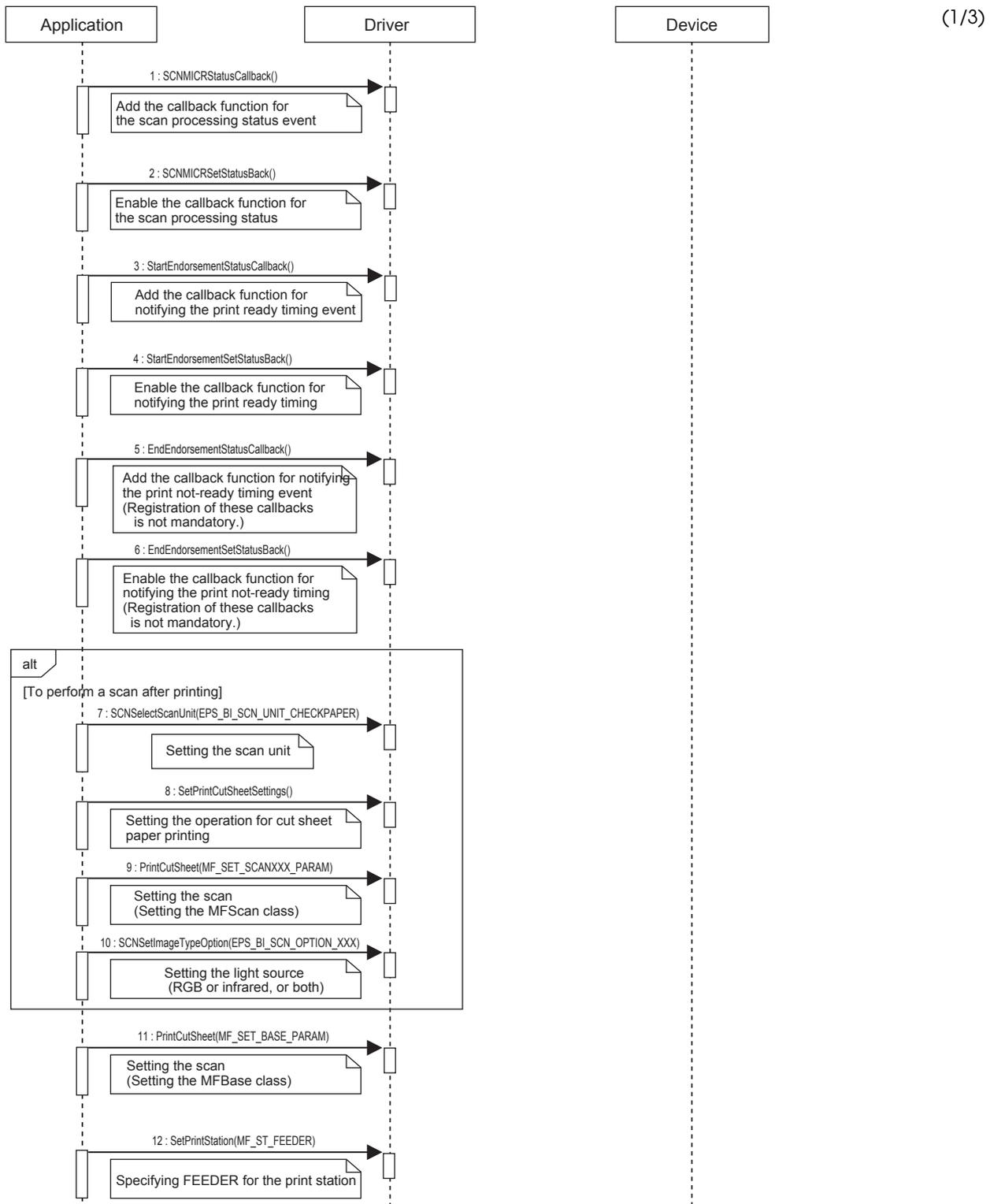
## Continuous printing sequence

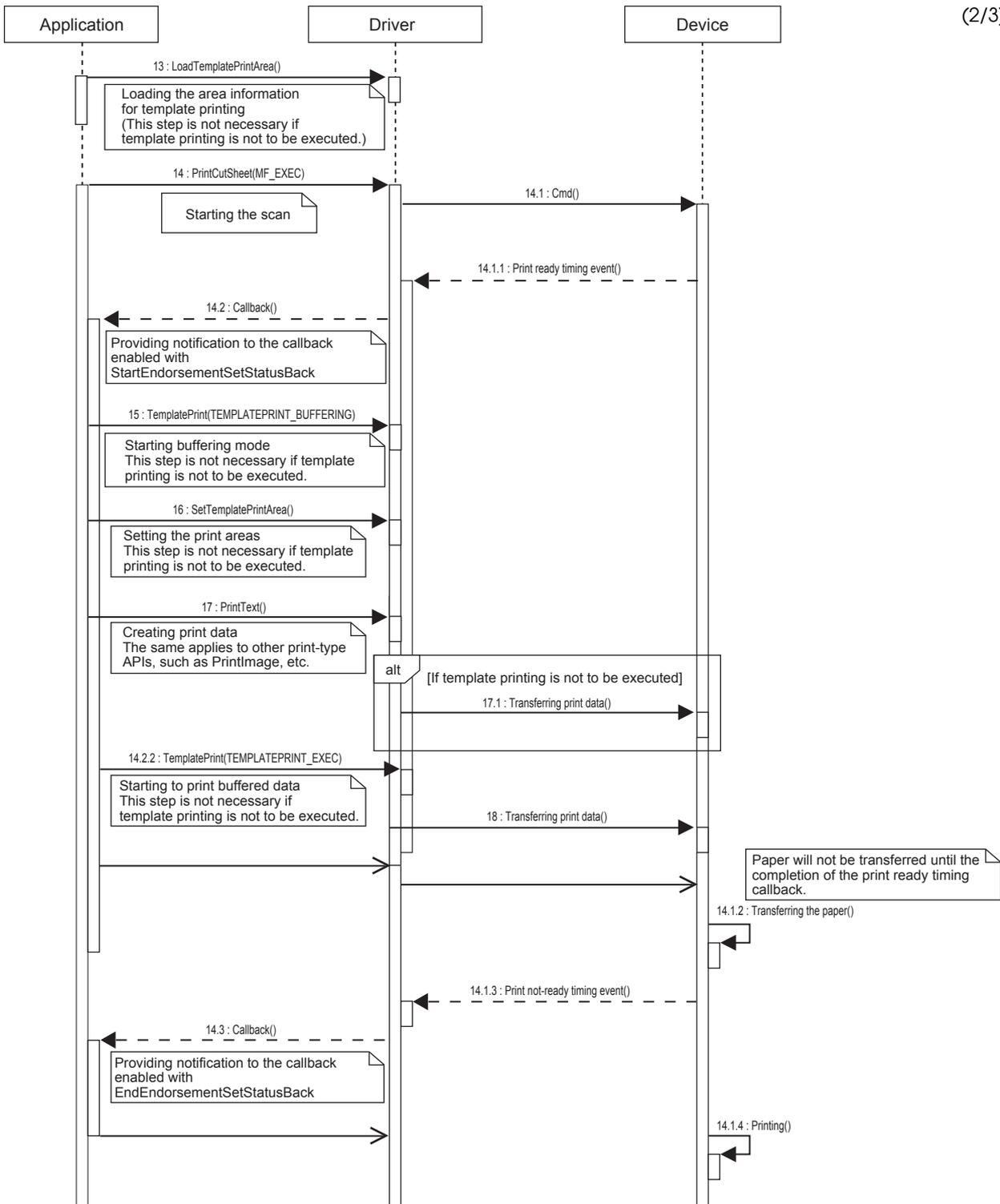
In this sequence, the same template is used for printing, with only the content changed for each sheet printed.

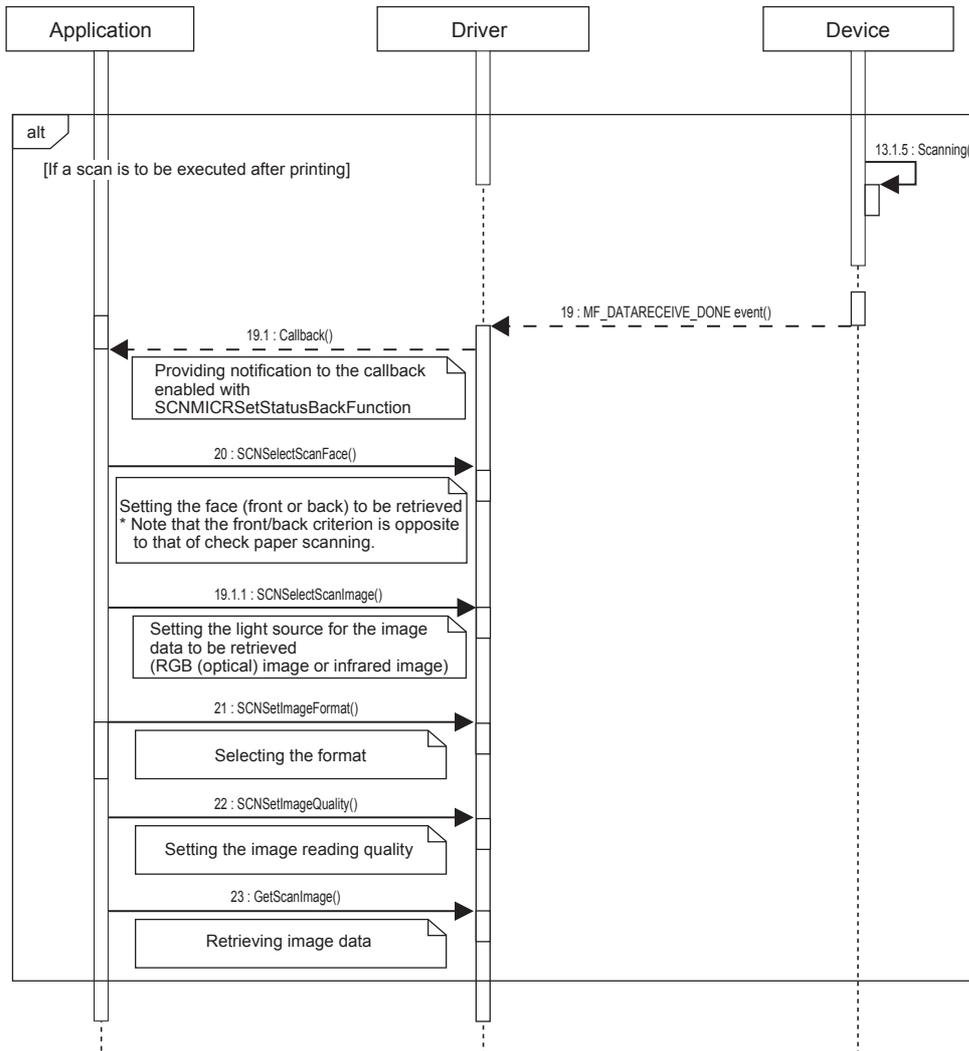


## Cut sheet paper printing

This is the process for printing and scanning a cut sheet, and then retrieving the scan results (image data).



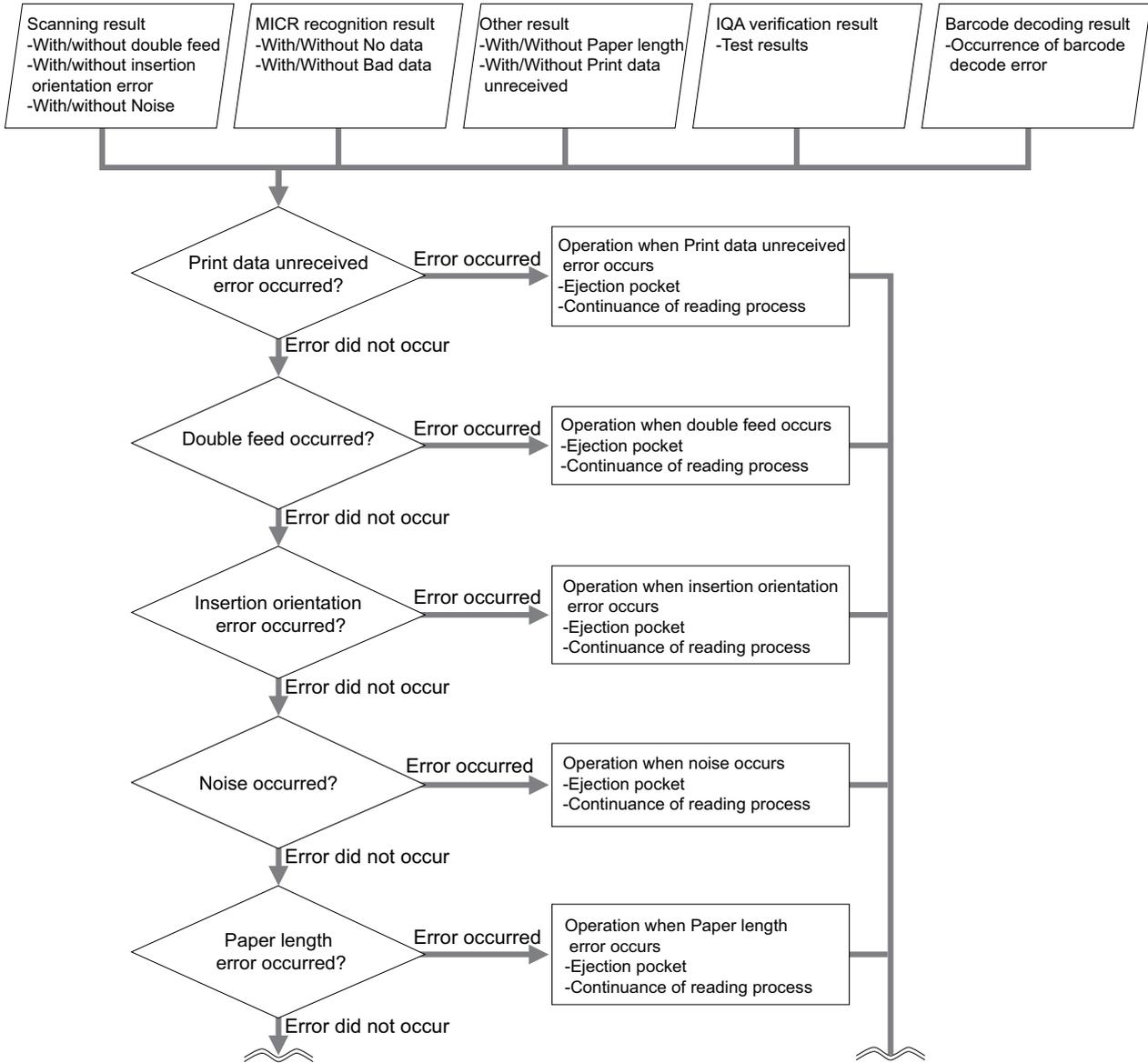




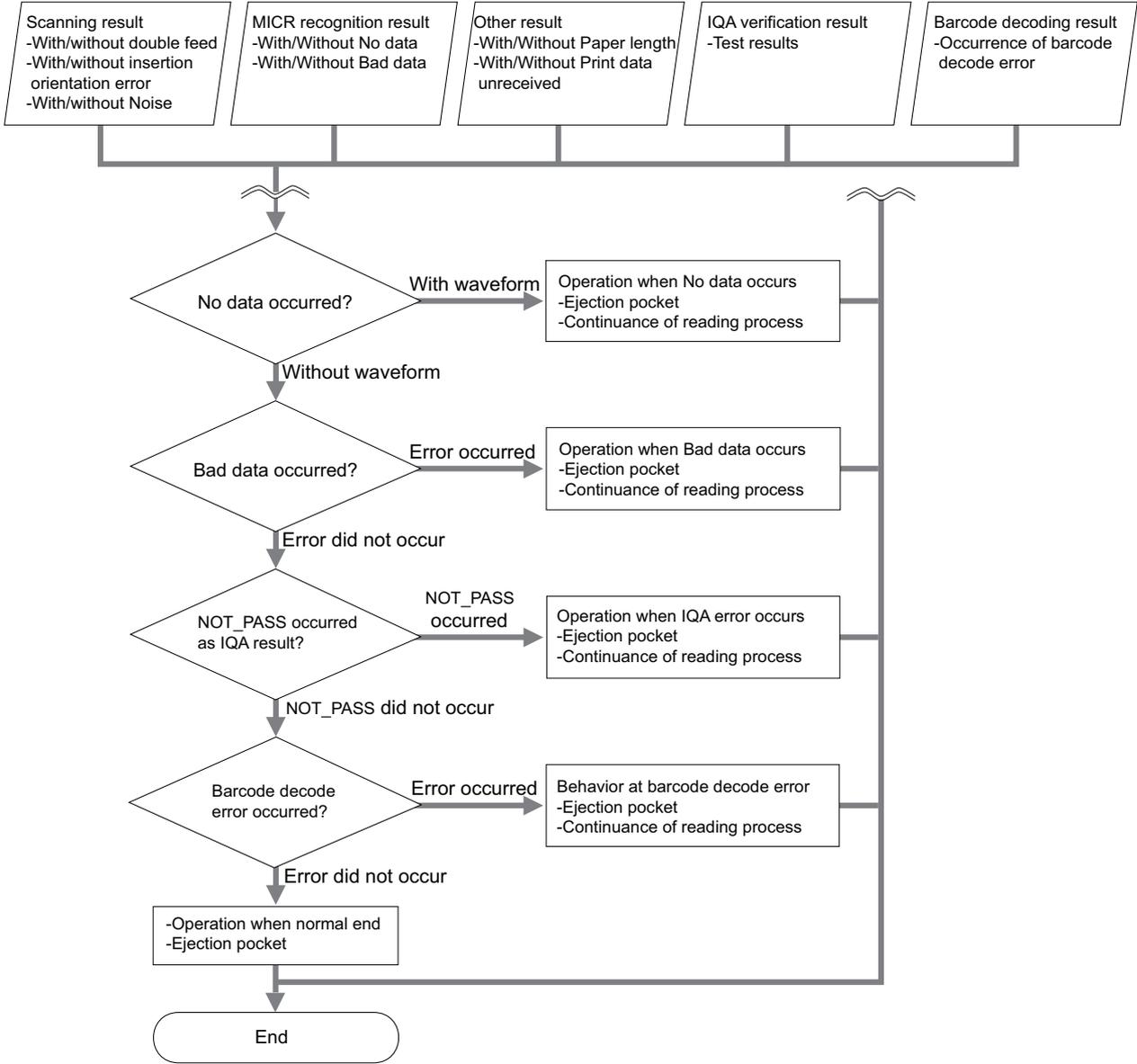
# Priority for error detection

Error detection (Double feed/Insertion orientation error/Noise/No data/Bad data/Paper length/Print data unreceived/IQA/ Barcode decode error) and operation priorities when an error occurs are shown below.

(1/2)



(2/2)



## Actions to be taken in the event of errors

There are two types of errors that may occur in the TM-S9000/S2000 .NET API: the error notified by the device status and the error that occurs when the TM-S9000/S2000 .NET API are called. A description of errors as well as solutions for fixing these errors is provided below. Please solve these errors in your own applications.

### Device status

The errors below occur when the device status is retrieved.

\* TM-S9000 only.

Macro Definition (Constant)	Cause
ASB_NO_RESPONSE	No response.
ASB_PRINT_SUCCESS	Printing completed.
ASB_DRAWER_KICK *	Drawer kick connector 3rd pin became high.
ASB_OFF_LINE	The device is in an off-line state.
ASB_MAIN_POCKET_NEAR_FULL	The main pocket is near full.
ASB_COVER_OPEN	The device cover is open.
ASB_PAPER_FEED *	Feeding the paper using the paper feed switch.
ASB_SUB_POCKET_NEAR_FULL	The sub pocket is near full.
ASB_WAIT_PEPRT_EJECT	Waiting for removal of a slip paper.
ASB_PANEL_SWITCH *	Paper feed switch is on.
ASB_MECHANICAL_ERR	A recoverable error occurred.
ASB_AUTOCUTTER_ERR *	An auto cutter error occurred.
ASB_UNRECOVER_ERR	An unrecoverable error occurred.
ASB_AUTORECOVER_ERR	An automatic recovery error occurred.
ASB_NOT_CARD_INSERT	No card is inserted in the PhotoID detector.
ASB_RECEIPT_NEAR_END *	No paper in the roll paper near end detector
ASB_MAIN_NEAR_FULL	The main pocket is not near full.
ASB_EJECT_SENSOR_NO_PAPER	No paper in the ejection sensor.
ASB_RECEIPT_END *	No paper in the roll paper end detector
ASB_SUB_NEAR_FULL	The sub pocket is not near full.
ASB_PAPER_INTERMEDIATE	No paper in the intermediate detector
ASB_SLIP_TOF	No paper in the TOF detector
ASB_ASF_PAPER	No paper in the ASF detector
ASB_SLIP_SELECTED	The cut sheet paper is not selected.
ASB_PRINT_SLIP	The cut sheet paper cannot be printed.
ASB_STAMP_EXIST	No stamp.
ASB_VALIDATION_SELECTED	The validation is not selected.
ASB_PRINT_VALIDATION	The validation cannot be printed.
ASB_WAIT_INSERT	Waiting for insertion of cut sheet paper.
ASB_SLIP_PAPER_SIZE	No paper in the paper length detector.
ASB_VALIDATION_NO_PAPER	No paper in the validation detector.
ASB_FRANKING_SENSOR	No paper in the franking detector.

## Return value

These errors (return values of the TM-S9000/S2000 .NET API) occur when the TM-S9000/S2000 .NET API are called. The content of the errors differ depending on the TM-S9000/S2000 .NET API.

Macro Definition (Constant)	Value	Cause	Solution
SUCCESS	0	Processing was completed successfully.	-
ERR_TYPE	-10	The nType parameter is incorrect.	Specify a correct constant that is defined in the header file.
ERR_OPENED	-20	The port is being used by other applications.	Terminate other applications.
ERR_NO_PRINTER	-30	The specified device cannot be found.	Check the connected status of the device. (Power status of the device, cable connection, etc.)
ERR_NO_TARGET	-40	An unsupported USB driver is used.	Reinstall the TM-S9000/S2000 driver.
ERR_NO_MEMORY	-50	Insufficient memory.	Terminate other applications or upgrade the memory.
ERR_HANDLE	-60	The handle value specifying the device is invalid.	Check if the handle value has been retrieved with OpenMonPrinter.
ERR_TIMEOUT	-70	A time-out error occurred.	If a time-out error occurs repeatedly, check the device status.
ERR_ACCESS	-80	The API did not complete successfully.	Check the device status. (Power status of the device, cable connection, etc.)
ERR_PARAM	-90	A parameter error occurred.	Check if the API parameter value or the value of the member of the class is correct.
ERR_NOT_SUPPORT	-100	The API feature called is not available.	<ul style="list-style-type: none"> <li>Check the API or the parameter called.</li> <li>If you are using the advanced scanner feature API, check if ESCNEnable has been called.</li> </ul>
ERR_OFFLINE	-110	The device is in an off-line state.	Check the device status.
ERR_NOT_EPSON	-120	A device other than an EPSON device is being used.	Check if an Epson device is being used.
ERR_WITHOUT_CB	-130	The CALLBACK function is not registered.	Call either SCNMICRSetStatusBackFunction or SCNMICRSetStatusBackWnd.
ERR_BUFFER_OVER_FLOW	-140	There is not enough memory available.	Increase the available memory size.
ERR_ENABLE	-160	OpenMonPrinter has already been called.	Execute CloseMonPrinter and then call OpenMonPrinter again.

Macro Definition (Constant)	Value	Cause	Solution
ERR_DISK_FULL	-170	There is not enough free disk space.	Review the operating environment.
ERR_NO_IMAGE	-180	The image file cannot be found.	Check if the image file exists at the specified path.
ERR_ENTRY_OVER	-190	The maximum number of entries that can be registered has already been reached.	-
ERR_CROPAREAID	-200	The specified CropArea does not exist.	Set DefineCropArea and then call DefineCropArea again.
ERR_EXIST	-210	The specified data already exists.	Change the parameter or call ESCNClearImage.
ERR_NOT_FOUND	-220	The specified data or module cannot be found.	<ul style="list-style-type: none"> <li>• If the data cannot be found, check the parameter of the executed API.</li> <li>• If the module cannot be found, reinstall the TM-S9000/S2000 driver.</li> </ul>
ERR_IMAGE_FILEOPEN	-230	An attempt to open a file failed.	Check if the file is not being used by other applications.
ERR_IMAGE_UNKNOWNFORMAT	-240	The file format is either invalid or not supported.	Check if the file can be opened successfully with other applications.
ERR_IMAGE_FAILED	-250	An attempt to save an image file failed.	Review the operating environment.
ERR_IMAGE_FILEREAD	-290	An attempt to load an image file failed.	Check if the file is located at the save destination.
ERR_PAPERINSERT_TIMEOUT	-300	An attempt to insert paper failed.	Check if the paper is set correctly in the device.
ERR_EXEC_FUNCTION	-310	The processing invoked by other API is being executed.	Wait until the processing completes and then call the API again.
ERR_RESET	-400	The device is unavailable because it is currently being reset.	Wait for a while and then call the API again.
ERR_ABORT	-430	The scan processing was cancelled.	Determine whether or not to execute scan processing again.
ERR_MICR	-440	An attempt to read MICR data failed.	Execute scan processing again.
ERR_SCAN	-450	An attempt to read image data failed.	Execute scan processing again.
ERR_NOT_EXEC	-470	The processing specified with the API was not executed.	<ul style="list-style-type: none"> <li>• Call the API again.</li> <li>• Check if API were called in correct order.</li> </ul>
ERR_DATA_INVALID	-480	The area information either did not exist or contained invalid data.	Check the area information file.
ERR_SIZE	-1000	The size is not set, or the value is invalid.	Check the value specified for the size and call the API again.

Macro Definition (Constant)	Value	Cause	Solution
ERR_PAPER_PILED	-1010	A double feed error occurred.	Execute scan processing again.
ERR_PAPER_JAM	-1020	A paper jam error occurred.	Remove the paper.
ERR_COVER_OPEN	-1030	Cover is open.	Close the cover.
ERR_MICR_NODATA	-1040	An attempt to detect magnetic waveform data failed.	Execute scan processing again.
ERR_MICR_BADDATA	-1050	An unanalyzable character was detected.	Execute scan processing again.
ERR_MICR_PARSE	-1060	An attempt to parse MICR data failed.	Execute scan processing again.
ERR_MICR_NOISE	-1070	A noise error was detected.	Execute scan processing again.
ERR_PAPER_EXIST	-1090	Paper exists in the path.	Remove the paper.
ERR_PAPER_INSERT	-1100	An attempt to insert paper failed.	Set the paper correctly in the device.
ERR_LESS_CHECKS	-1110	An attempt to scan a specified number of sheets failed.	Either change the specified number of sheets or increase the number of sheets to be loaded.
ERR_SCN_IQA	-1120	NOT_PASS is detected at the IQA validation.	Load GetIQAResult, check the test results, and determine whether to scan again.
ERR_BARCODE_NODATA	-1130	An attempt to detect a barcode failed.	-
ERR_IMAGE_FILEREMOVE	-1170	An attempt to delete an image file failed.	Check if the file is not being used by other applications.
ERR_PRINT_DATA_LENGTH_EXCEED	-1180	The printing content exceeds printable area.	The printing content exceeds printable area. Use an appropriate paper size.
ERR_PRINT_DATA_UNRECEIVE	-1190	Printing data not received error generated	Review the operating environment, or set EndorsePrintMode for the MFProcess class to MF_ENDORSEPRINT_MODE_DATAWAITING and execute again.
ERR_INK_STATUS *	-1200	No ink cartridge or ink is low.	Install a new ink cartridge.
ERR_UNKNOWN	-9999	A module cannot be loaded.	Reinstall the driver.

\* Supported in Ver.2.10 or later

# Sample Program

Programming with the TM-S9000/S2000 API is described using 8 functional level sample programs.

Sample Program	Description
Step1	This is a sample program for opening and closing devices, and scanning.
Step2	In addition to Step 1, this sample program allows you to select to scan either a check sheet or a card, and then retrieve and display the scanned images.
Step3	In addition to Step 2, this sample program retrieves and displays MICR data. It also cleans the MICR.
Step4	In addition to Step 3, this sample program performs physical endorsement/ electronic endorsement.
Step5	In addition to Step 3, this sample program processes cashier's checks. It also prints cut sheets/roll paper (TM-S9000 only).
Step6	In addition to Step 3, this sample program retrieves and displays OCR-AB recognition results. It also sounds an alarm.
Step7	In addition to Step 3, this sample program processes IQA and displays the results. It also allows the selection of scanning mode (High Speed Mode/Confirmation Mode/Waterfall).
Step8	In addition to Step 3, this sample program retrieves and displays barcode decode results, and detects errors. It also retrieves device status and device information.



Sample programs are designed so that the screens and components are separate, and therefore they can easily be copied and added to customers' programs.

# TM-S9000/S2000 API Reference

This chapter provides a description of the TM-S9000/S2000 API and their syntax.



This chapter describes the syntax of APIs and classes. For more information about individual APIs, refer to the [API Comparison Chart \(Alphabetical Order\)](#) (p.47) and the TM-S9000/S2000 API Reference Card for Win32/64.

## API Comparison Chart (Alphabetical Order)

Win32/64 API	.NET API
int BiAutoCutRollPaper(int nHandle, int nMethod)	ErrorCode AutoCutRollPaper (AutoCufType autocuttype)
int BiBufferedPrint(int nHandle, DWORD dwFunction)	ErrorCode BufferedPrint (PrintBuffer function)
int BiCancelError(int nHandle)	ErrorCode CancelError()
int BiCancelInkStatusBack(int nHandle)	ErrorCode CancelInkStatusBack()
int BiCancelStatusBack(int nHandle)	ErrorCode CancelStatusBack()
int BiClearTemplatePrintData (int nHandle, unsigned long ulTransactionNumber)	ErrorCode ClearTemplatePrintData (uint transactionNumber)
int BiCloseMonPrinter(int nHandle)	ErrorCode CloseMonPrinter()
int BiConfirmBufferedData(int nHandle, LPDWORD lpdwBufferedCount)	-
int BiDecodeBarcode(int nHandle, LPCSTR szFileName, LPMF_BARCODE ptBarcode)	ErrorCode DecodeBarcode (String szFileName, MFBarcode mfBarcode)
int BiDecodeBarcodeMemory(int nHandle, LPBYTE lpImageBuffer, DWORD dwBufferSize, LPMF_BARCODE ptBarcode)	ErrorCode DecodeBarcodeMemory (Bitmap objBitmap, MFBarcode mfBarcode)
int BiEndEndorsementCancelStatusBack(int nHandle)	ErrorCode EndEndorsementCancelStatusBack()
int BiEndEndorsementSetStatusBackFunction(int nHandle, int (CALLBACK EXPORT* pEndorseCB) ( unsigned long ulTransactionNumber, char* pszPortName ))	EndEndorsementStatusCallback
int BiEndEndorsementSetStatusBackWnd (int nHandle, HWND hWnd, unsigned long* pulTransactionNumber)	
int BiESCNClearImage (int nHandle, BYTE bFlag, DWORD dwFileIndex, LPSTR pFileID, LPSTR pImageTagData)	ErrorCode ESCNClearImage (ClearImageFlag flag, int fileIndex, String fileID, String imageTagData)
int BiESCNDefineCropArea (int nHandle, BYTE bCropAreaID, WORD wStartX, WORD wStartY, WORD wEndX, WORD wEndY)	ErrorCode ESCNDefineCropArea (byte areaNo, Rectangle area)
int BiESCNEnable(BYTE bStoreType)	static ErrorCode ESCNEnable (Storage storage)
int BiESCNGetAutoSize(int nHandle, LPBYTE pCapAutoSize)	ErrorCode ESCNGetAutoSize (out AutoSize autoSize)
	ESCNAutoSize

Win32/64 API	.NET API
int BiESCNGetCutSize (int nHandle, LPWORD pCutSize)	ErrorCode ESCNGetCutSize (out int cutSize) ESCNCutSize
int BiESCNGetDeSkew (int nHandle, LPWORD lpwAngle)	ErrorCode ESCNGetDeSkew (out ushort lpwAngle) ESCNDeSkew
int BiESCNGetDocumentSize (int nHandle, LPWORD pDocumentWidth, LPWORD pDocumentHeight)	ErrorCode ESCNGetDocumentSize (out int documentWidth, out int documentHeight) ESCNDocumentSize
int BiESCNGetMaxCropAreas (int nHandle, LPBYTE pMaxCropAreas)	ErrorCode ESCNGetMaxCropAreas (out int count)
int BiESCNGetRemainingImages (int nHandle, LPBYTE pRemainingImages)	ErrorCode ESCNGetRemainingImages (out int count)
int BiESCNGetRotate (int nHandle, LPBYTE pCapRotate)	ErrorCode ESCNGetRotate (out Rotate rotate) ESCNRotate
int BiESCNRetrievImage (int nHandle, DWORD dwFileIndex, LPSTR pFileID, LPSTR plmageTagData, LPDWORD plmageSize, LPBYTE plmageData)	ErrorCode ESCNRetrievImage (int fileIndex, String fileID, String imageTagData, out byte[] data) ErrorCode ESCNRetrievImage (int fileIndex, String fileID, String imageTagData, out Image image)
int BiESCNSetAutoSize(int nHandle, BYTE bCapAutoSize)	ErrorCode ESCNSetAutoSize (AutoSize autoSize) ESCNAutoSize
int BiESCNSetCutSize (int nHandle, WORD wCutSize)	ErrorCode ESCNSetCutSize (int cutSize) ESCNCutSize
int BiESCNSetDeSkew (int nHandle, WORD wAngle)	ErrorCode ESCNSetDeSkew (ushort wAngle) ESCNDeSkew
int BiESCNSetDocumentSize (int nHandle, WORD wDocumentWidth, WORD wDocumentHeight)	ErrorCode ESCNSetDocumentSize (int documentWidth, int documentHeight) ESCNDocumentSize
int BiESCNSetRotate (int nHandle, BYTE bCapRotate)	ErrorCode ESCNSetRotate (Rotate rotate) ESCNRotate
int BiESCNStoreImage (int nHandle, DWORD dwFileIndex, LPSTR pFileID, LPSTR plmageTagData, BYTE bCropAreaID)	ErrorCode ESCNStoreImage (int fileIndex, String fileID, String imageTagData, byte cropAreaID)
int BiGetBarcodeData (int nHandle, DWORD dwTransactionNumber, LPMF_BARCODE ptBarcode)	ErrorCode GetBarcodeData (int dwTransactionNumber, MFBarcode mfBarcode)
int BiGetCounter (int nHandle, WORD readno, LPDWORD readcounter)	ErrorCode GetCounter (byte counter, out int value) ErrorCode GetCounter (CounterIndex counter, bool cumulative, out int value)
int BiGetInkStatus (int nHandle, LPBYTE pStatus)	InkStatus
int BiGetIQAResult (int nHandle, DWORD dwTransactionNumber, LPMF_IQA_RESULT lpResult)	ErrorCode GetIqaResult (int dwTransactionNumber, MFIqaResult pResult)
int BiGetMicrText(int nHandle, DWORD dwTransactionNumber, LPMF_MICR ptMicr)	ErrorCode GetMicrText (int transactionNumber, MFMicr mfMicr)

Win32/64 API	.NET API
int BiGetOcrABText (int nHandle, DWORD dwTransactionNumber , BYTE blmImageSource, LPCSTR szFileName , LPMF_OCR_AB ptOcrAB)	ErrorCode GetOcrABText (uint dwTransactionNumber, OcrImageSource elmImageSource, String szFileName, MFOcrAB mfOcrAB)
int BiGetOfflineCode (int nHandle, LPBYTE lpOfflinecode)	OfflineCode
int BiGetOfflineCodeByIndex (int nHandle, int nIndex, LPBYTE lpOfflinecode)	ErrorCode GetOfflineCodeByIndex (int nIndex, out byte lpbOfflinecode)
int BiGetPrintAlignment (int nHandle, LPDWORD lpdwAlignment)	ErrorCode GetPrintAlignment (out Alignment alignment) PrintAlignment
int BiGetPrintControl (int nHandle, LPBYTE lpbSpeed , LPBYTE lpbDirection)	ErrorCode GetPrintControl (out PrintSpeed speed, out PrintDirection direction)
int BiGetPrintCutSheetSettings (int nHandle, unsigned char* pucNumber , unsigned char* pucScan)	ErrorCode GetPrintCutSheetSettings (out int Number, out ScanEnable Scan)
int BiGetPrintImageMethod (int nHandle, int* pnMethod)	ErrorCode GetPrintImageMethod (out PrintImageType method)
int BiGetPrintPosition (int nHandle, LPWORD lpwHorizontal , LPWORD lpwVertical)	ErrorCode GetPrintPosition (out int horizontal, out int vertical) PrintPosition
int BiGetPrintSize (int nHandle, LPWORD lpwWidth , LPWORD lpwHeight)	ErrorCode GetPrintSize (out int width, out int height) PrintSize
int BiGetPrintStation (int nHandle, LPWORD lpwStation)	ErrorCode GetPrintStation (out PrintingStation station) PrintStation
int BiGetPrnCapability (int nHandle, BYTE prnID, LPBYTE pBuffSize , LPBYTE pBuff)	ErrorCode GetPrnCapability (byte printerID, out byte[] data)
int BiGetRealStatus (int nHandle, LPDWORD lpStatus)	ErrorCode GetRealStatus (out ASB asb)
int BiGetScanImage (int nHandle, DWORD dwTransactionNumber , LPMF_SCAN ptScan)	ErrorCode GetScanImage (int transactionNumber, MFScan mfScan)
int BiGetScanImageSize (int nHandle, DWORD dwTransactionNumber , LPMF_SCAN ptScan)	-
int BiGetStatus (int nHandle, LPDWORD lpStatus)	Status
int BiGetTransactionNumber (int nHandle, LPDWORD lpdwTransactionNumber)	ErrorCode GetTransactionNumber (out int transactionNumber) TransactionNumber
int BiGetType (int nHandle, LPBYTE typeID, LPBYTE font , LPBYTE exrom, LPBYTE special)	ErrorCode GetType (out byte typeid, out byte font, out byte exrom, out byte euspecial)

Win32/64 API	.NET API
int BiGetVersion (int nDriverType, int nType , LPVERSION_INFO lpVersion)	ErrorCode GetVersion (DriverType eDriverType, VersionType eType, MFVersion ptVersion)
int BiInkHeadCleaning (int nHandle)	ErrorCode InkHeadCleaning()
int BiInsertValidation (int nHandle, DWORD dwTimeout)	ErrorCode InsertValidation (int timeout)
int BiLoadAPISettings (int nHandle, char* pszSettingsFilePath)	ErrorCode LoadAPISettings (String SettingsFilePath)
int BiLoadTemplatePrintArea (int nHandle, char* pszFilePath, int* pnLoadNum)	ErrorCode LoadTemplatePrintArea (String FilePath, out int LoadNum)
int BiMICRCleaning (int nHandle)	ErrorCode MICRCleaning()
int BiMICRClearSpaces (int nHandle, BYTE bClearSpace)	ErrorCode MICRClearSpaces (RemoveSpace removeSpace)
int BiMICRGetStatus (int nHandle, LPBYTE pStatus)	ErrorCode MICRGetStatus (out byte status)
int BiMICRSelectDataHandling (int nHandle, BYTE charSelect, BYTE detailSelect , BYTE errorSelect)	ErrorCode MICRSelectDataHandling (CharSelect charSel, DetailSelect detailSel, ErrorSelect errorSel)
int BiOpenDrawer (int nHandle, BYTE byDrawer, BYTE byPulse)	ErrorCode OpenDrawer (Drawer drawer, OpenPulse pulse)
int BiOpenMonPrinter (int nType, LPSTR pName)	ErrorCode OpenMonPrinter (OpenType type, String name)
int BiPrintBarCode (int nHandle, LPBYTE pbyData , DWORD dwDataLength, DWORD dwSymbol , DWORD dwTextPosition)	ErrorCode PrintBarCode (byte[] data, Symbol symbol, TextPosition textPosition) ErrorCode PrintBarCode (String data, Symbol symbol, TextPosition textPosition)
int BiPrintCutSheet (int nHandle, void* lpvStruct , unsigned short usFunction)	ErrorCode PrintCutSheet (FunctionType functionType) ErrorCode PrintCutSheet (MF mf, FunctionType functionType)
int BiPrintImage (int nHandle, LPSTR pFileName)	ErrorCode PrintImage (String filename)
int BiPrintMemoryImage (int nHandle, LPBYTE lpblmageData , DWORD dwDataSize)	ErrorCode PrintMemoryImage (Bitmap objImage)
int BiPrintMultipleToneImage (int nHandle, char szKeyCode1, char szKeyCode2)	ErrorCode PrintMultipleToneImage (byte bKeyCode1, byte bKeyCode2)
int BiPrintText (int nHandle, LPSTR szText , MF_DECORATE tDecorate)	ErrorCode PrintText (String text, MFDecorate decorate)
int BiRemoveValidation (int nHandle, DWORD dwTimeout)	ErrorCode RemoveValidation (int timeout)
int BiResetCounter (int nHandle, WORD writeno)	ErrorCode ResetCounter (CounterIndex counter) ErrorCode ResetCounter (byte counter)
int BiResetPrinter (int nHandle)	ErrorCode ResetPrinter()
int BiRingBuzzer (int nHandle, BYTE bTone, BYTE bCount , WORD wOnTime, WORD wOffTime)	ErrorCode RingBuzzer (MfBuzzerTone bTone, byte bCount, ushort wOnTime, ushort wOffTime)
int BiSCNDeleteCroppingArea (int nHandle, BYTE AreaNo)	ErrorCode SCNDeleteCroppingArea (byte areaNo)

Win32/64 API	.NET API
int BiGetInkStatus ( int nHandle, LPBYTE pStatus)	ErrorCode SCNGetClampStatus (out byte status)
int BiSCNGetCroppingArea (int nHandle, LPWORD pBuffSize, LPBYTE pBuff)	ErrorCode SCNGetCroppingArea (out byte[] areas)
	ErrorCode SCNGetCroppingArea (out Hashtable areas)
int BiSCNGetImageAdjustment (int nHandle, int* pnBrightness, int* pnContrast , unsigned long* pulGamma)	ErrorCode SCNGetImageAdjustment (out Brightness brightness, out Contrast contrast, out Gamma gamma)
int BiSCNGetImageFormat (int nHandle, LPBYTE pFormat)	ErrorCode SCNGetImageFormat (out Format format)
int BiSCNGetImageQuality (int nHandle, BYTE bColorDepth, char bThreshold , BYTE bColor, BYTE bExOption)	ErrorCode SCNGetImageQuality (out ColorDepth colorDepth, out double threshold, out Color color, out ExOption exOption)
int BiSCNGetImageTypeOption (int nHandle, unsigned long* pulOption)	ErrorCode SCNGetImageTypeOption (out ImageTypeOption imagetypeoption)
int BiSCNGetScanArea (int nHandle, LPBYTE pStartX, LPBYTE pStartY , LPBYTE pEndX, LPBYTE pEndY)	ErrorCode SCNGetScanArea (out Rectangle area)
	ErrorCode SCNGetScanArea (out int startX, out int startY, out int endX, out int endY)
int BiSCNMICRCancelFunction (int nHandle, WORD wEjectType)	ErrorCode SCNMICRCancelFunction (MfEjectType ejectType)
int BiSCNMICRCancelStatusBack (int nHandle)	ErrorCode SCNMICRCancelStatusBack()
int BiSCNMICRFunction (int nHandle, LPVOID lpvStruct, WORD wFunction)	ErrorCode SCNMICRFunction (FunctionType functionType)
	ErrorCode SCNMICRFunction (MF mf, FunctionType functionType)
int BiSCNMICRFunctionContinuously (int nHandle, LPVOID lpvStruct, WORD wFunction)	ErrorCode SCNMICRFunctionContinuously (FunctionType functionType)
	ErrorCode SCNMICRFunctionContinuously (MF mf, FunctionType functionType)
int BiSCNMICRFunctionPostPrint (int nHandle, LPVOID lpvStruct, WORD wFunction)	ErrorCode SCNMICRFunctionPostPrint (FunctionType functionType)
	ErrorCode SCNMICRFunctionPostPrint (MF mf, FunctionType functionType)
int BiSCNMICRSetStatusBackFunction (int nHandle, int (CALLBACK EXPORT* pScnMicrCB) (DWORD dwTransactionNumber , WORD wMainStatus, WORD wSubStatus , LPSTR lpcPortName))	SCNMICRStatusCallback
int BiSCNMICRSetStatusBackWnd (int nHandle, long hWnd , LPDWORD lpdwTransactionNumber , LPWORD lpwMainStatus, LPWORD lpwSubStatus)	
int BiSCNMICRSetStatusBackWndEx (int nHandle, HWND hWnd , LPDWORD pdwTransactionNumber , LPWORD pwMainStatus, LPWORD pwSubStatus)	

Win32/64 API	.NET API
int BiSCNPrintMemoryImage (int nHandle, unsigned long ulTransactionNumber , LPBYTE lpblImageData, DWORD dwDataSize)	ErrorCode SCNPrintMemoryImage (uint transactionNumber, Bitmap objImage)
int BiSCNPrintText (int nHandle, unsigned long ulTransactionNumber , LPSTR szText, MF_DECORATE* pDecorate)	ErrorCode SCNPrintText (uint transactionNumber, String text , MFDecorate decorate)
int BiSCNSelectScanFace (int nHandle, BYTE bFace)	ErrorCode SCNSelectScanFace (ScanSide scanFace)
int BiSCNSelectScanImage (int nHandle, unsigned char uclImage)	ErrorCode SCNSelectScanImage (ImageType imagetype)
int BiSCNSelectScanUnit (int nHandle, BYTE bSelectUnit)	ErrorCode SCNSelectScanUnit (ScanUnit scanUnit)
int BiSCNSetCroppingArea (int nHandle, BYTE bAreaNo, BYTE bStartX , BYTE bStartY, BYTE bEndX, BYTE bEndY)	ErrorCode SCNSetCroppingArea (byte areaNo, int startX, int startY, int endX, int endY) ErrorCode SCNSetCroppingArea (byte areaNo, Rectangle area)
int BiSCNSetImageAdjustment (int nHandle, int nBrightness, int nContrast , unsigned long ulGamma)	ErrorCode SCNSetImageAdjustment (Brightness brightness, Contrast contrast, Gamma gamma)
int BiSCNSetImageFormat (int nHandle, BYTE bFormat)	ErrorCode SCNSetImageFormat (Format format)
int BiSCNSetImageQuality (int nHandle, BYTE bColorDepth, char bThreshold , BYTE bColor, BYTE bExOption)	ErrorCode SCNSetImageQuality (ColorDepth colorDepth, double threshold, Color color, ExOption exOption)
int BiSCNSetImageTypeOption (int nHandle, unsigned long ulOption)	ErrorCode SCNSetImageTypeOption (ImageTypeOption imagetypeoption)
int BiSCNSetScanArea (int nHandle, BYTE bStartX, BYTE bStartY , BYTE bEndX, BYTE bEndY)	ErrorCode SCNSetScanArea (Rectangle area) ErrorCode SCNSetScanArea (int startX, int startY, int endX, int endY)
int BiSelectErrorEjectAtContinuously (int nHandle , DWORD dwError , DWORD dwSetting)	ErrorCode SelectErrorEjectAtContinuously (ErrorEjectError error, ErrorEjectSetting setting)
int BiSelectJamDetect (int nHandle, DWORD dwJamDetect)	ErrorCode SelectJamDetect (JamDetect jamDetect)
int BiSetBehaviorToScnResult (int nHandle, BYTE bEject, BYTE bStamp , BYTE bNextCheck)	ErrorCode SetBehaviorToScnResult (MfEject bEject, MfStamp bStamp, MfProcessContinue bNextCheck)
int BiSetConfigure (int nHandle, unsigned long ulConfigID , unsigned long ulConfigType)	ErrorCode SetConfigure (ConfigID configID, ConfigType configType)
int BiSetEndorseDirection (int nHandle, BYTE bDirection)	ErrorCode SetEndorseDirection (ElectricEndorseDirection eDirection)

Win32/64 API	.NET API
int BiSetInkStatusBackFunction (int nHandle, int (CALLBACK EXPORT *pStatusCB) (WORD wStatus))	InkStatusCallback
int BiSetInkStatusBackWnd (int nHandle, long hWnd, LPDWORD lpStatus)	
int BiSetStatusBackWndEx (int nHandle, HWND hWnd, LPDWORD pdwStatus)	
int BiSetInkStatusBackFunctionEx (int nHandle, int (CALLBACK EXPORT *pStatusCB) (WORD wStatus, LPSTR lpcPortName))	InkStatusCallbackEx
int BiSetMonInterval (int nHandle, WORD wNoPrnInterval , WORD wPrnInterval)	ErrorCode SetMonInterval (int noPrnInterval, int prnInterval)
int BiSetNumberOfDocuments (int nHandle, BYTE bNumber)	ErrorCode SetNumberOfDocuments (byte bNumber)
int BiSetOcrABAreaOrigin (int nHandle, BYTE bOrigin)	ErrorCode SetOcrABAreaOrigin (OcrOrigin eOcrOrigin)
int BiSetPaperThickness (int nHandle, WORD wThreshold)	ErrorCode SetPaperThickness (ushort wThreshold)
int BiSetPrintAlignment (int nHandle, DWORD dwAlignment)	ErrorCode SetPrintAlignment (Alignment alignment)
int BiSetPrintControl (int nHandle, BYTE bSpeed, BYTE bDirection)	ErrorCode SetPrintControl (PrintSpeed speed, PrintDirection direction)
int BiSetPrintCutSheetSettings (int nHandle, unsigned char ucNumber , unsigned char ucScan)	ErrorCode SetPrintCutSheetSettings (byte Number, ScanEnable scan)
int BiSetPrintPosition (int nHandle, WORD wHorizontal , WORD wVertical)	ErrorCode SetPrintPosition (int horizontal, int vertical)
int BiSetPrintSize (int nHandle, WORD wWidth, WORD wHeight)	ErrorCode SetPrintSize (int width, int height)
int BiSetPrintStation (int nHandle, WORD wStation)	ErrorCode SetPrintStation (PrintingStation station)
int BiSetStatusBackFunction (int nHandle, int (CALLBACK EXPORT *pStatusCB) (DWORD dwStatus))	StatusCallback
int BiSetStatusBackWnd (int nHandle, long hWnd, LPDWORD lpStatus)	
int BiSetStatusBackWndEx (int nHandle, HWND hWnd, LPDWORD pdwStatus)	
int BiSetStatusBackFunctionEx (int nHandle, int (CALLBACK EXPORT *pStatusCB) (DWORD dwStatus, LPSTR lpcPortName))	StatusCallbackEx
int BiSetTemplatePrintArea (int nHandle, int nAreaSelectMode , char* pszAreaName, LPPRINTAREAINFO pstInfo , unsigned long ulTransactionNumber)	ErrorCode SetTemplatePrintArea (AreaSelectMode mode, String areaname , MFPrintAreaInfo info, int transactionNumber)

Win32/64 API	.NET API
int BiSetTransactionNumber (int nHandle, DWORD dwTransactionNumber)	ErrorCode SetTransactionNumber (int transactionNumber) TransactionNumber
int BiSetTransactionNumberWithIncremental (int nHandle, DWORD dwTransactionNumber , DWORD dwIncremental)	ErrorCode SetTransactionNumberWithIncrement (int number, int increment)
int BiSetWaterfallMode (int nHandle, BYTE bWaterfallMode)	ErrorCode SetWaterfallMode (WaterfallMode eWaterfallMode)
int BiStartEndorsementCancelStatusBack (int nHandle)	ErrorCode StartEndorsementCancelStatusBack()
int BiStartEndorsementSetStatusBackFunction (int nHandle, int (CALLBACK EXPORT* pEndorseCB) ( unsigned long ulTransactionNumber , char* pszPortName))	StartEndorsementStatusCallback
int BiStartEndorsementSetStatusBackWnd (int nHandle, HWND hWnd , unsigned long* pulTransactionNumber)	
int BiTemplatePrint (int nHandle, int nFunction)	ErrorCode TemplatePrint (TemplatePrintMode templateprintmode)
int BiUpdateEndorseText (int nHandle, LPSTR lpString(3) , DWORD, dwAttribute(3), WORD wFont(3) , WORD wFontSize(3))	ErrorCode UpdateEndorseText (String [] strings, int [] attributes, FontType [] fonts, FontScale [] fontSizes) ErrorCode UpdateEndorseText (MFPrint mfPrint)
	IsValid
	LastError
	MfDone
	ResetLastError
	TransactionIncrement

## Class list

The TM-S9000/S2000 Driver implements .NET Wrapper. (The TM-S9000/S2000 is controlled with .NET Wrapper from your .NET applications.) The .NET Wrapper has the following classes.

Class	Description	Page
MFDevice	Main class. The device is controlled by calling a property method defined in this class.	<a href="#">p. 59</a>
MFBBase	Class corresponding to the MF_BASE01 structure in Win32/64 API.	<a href="#">p. 201</a>
MFScan	Class corresponding to the MF_SCAN structure in Win32/64 API.	<a href="#">p. 212</a>
MFMicr	Class corresponding to the MF_MICR01 structure in Win32/64 API.	<a href="#">p. 205</a>
MFDeviceFont	Class used for the PrintText/SCNPrintText method.	<a href="#">p. 215</a>
MFTrueType	Class used for the PrintText/SCNPrintText method.	<a href="#">p. 218</a>
MFPrintAreaInfo	Class used for the SetTemplatePrintArea method.	<a href="#">p. 269</a>
MFProcess	Class corresponding to the MF_PROCESS01 structure in Win32/64 API.	<a href="#">p. 220</a>
MF_Iqa	Class corresponding to the MF_IQA structure in Win32/64 API.	<a href="#">p. 232</a>
MF_IqaResult	Class corresponding to the MF_IQA_RESULT structure in Win32/64 API.	<a href="#">p. 242</a>
MFBarcode	Class corresponding to the MF_BARCODE structure in Win32/64 API.	<a href="#">p. 252</a>
MFOcrAb	Class corresponding to the MF_OCR_AB structure in Win32/64 API.	<a href="#">p. 263</a>
MFVersion	Class corresponding to the VERSION_INFO structure in Win32/64 API.	<a href="#">p. 268</a>
MFPrint	Class corresponding to the MF_PRINT01 structure in Win32/64 API.	<a href="#">p. 271</a>

# API Index

## Basic APIs

API	Page	API	Page
OpenMonPrinter	<a href="#">p. 61</a>	CloseMonPrinter	<a href="#">p. 63</a>
SCNMICRCancelStatusBack	<a href="#">p. 63</a>	StartEndorsementCancelStatusBack	<a href="#">p. 64</a>
EndEndorsementCancelStatusBack	<a href="#">p. 64</a>	SCNMICRFunctionContinuously	<a href="#">p. 74</a>
SCNMICRFunctionPostPrint	<a href="#">p. 79</a>	PrintImage	<a href="#">p. 127</a>
SCNPrintMemoryImage	<a href="#">p. 131</a>	PrintText	<a href="#">p. 132</a>
SCNPrintText	<a href="#">p. 133</a>	TemplatePrint	<a href="#">p. 145</a>
PrintCutSheet	<a href="#">p. 140</a>	ClearTemplatePrintData	<a href="#">p. 149</a>
SCNSelectScanUnit	<a href="#">p. 69</a>	SCNSelectScanFace	<a href="#">p. 86</a>
SCNSetImageTypeOption	<a href="#">p. 70</a>	SetPrintStation	<a href="#">p. 114</a>
SetPrintCutSheetSettings	<a href="#">p. 139</a>	LoadTemplatePrintArea	<a href="#">p. 144</a>
SetTemplatePrintArea	<a href="#">p. 147</a>	SCNSetImageFormat	<a href="#">p. 89</a>
SCNSetImageQuality	<a href="#">p. 93</a>	SCNSelectScanImage	<a href="#">p. 95</a>
GetScanImage	<a href="#">p. 106</a>	GetMicrText	<a href="#">p. 104</a>
CancelStatusBack	<a href="#">p. 66</a>	CancelInkStatusBack	<a href="#">p. 66</a>
CancelError	<a href="#">p. 187</a>	ResetPrinter	<a href="#">p. 186</a>
SCNGetImageTypeOption	<a href="#">p. 73</a>	GetPrintStation	<a href="#">p. 111</a>
GetPrintCutSheetSettings	<a href="#">p. 138</a>	SCNGetImageFormat	<a href="#">p. 88</a>
SCNGetImageQuality	<a href="#">p. 92</a>	ESCNEnable	<a href="#">p. 156</a>
BufferedPrint	<a href="#">p. 116</a>	SetPrintPosition	<a href="#">p. 114</a>
SetPrintSize	<a href="#">p. 120</a>		

## API List (Alphabetical Order)

API	Page	API	Page
<b>A</b>			
AutoCutRollPaper	p. 143		
<b>B</b>			
BufferedPrint	p. 116		
<b>C</b>			
CancelError	p. 187	CancelLinkStatusBack	p. 66
CancelStatusBack	p. 66	ClearTemplatePrintData	p. 149
CloseMonPrinter	p. 63		
<b>D</b>			
DecodeBarcode	p. 108	DecodeBarcodeMemory	p. 109
<b>E</b>			
EndEndorsementCancelStatusBack	p. 64	ESCNClearImage	p. 170
ESCNDefineCropArea	p. 167	ESCNEnable	p. 156
ESCNGetAutoSize	p. 157	ESCNGetCutSize	p. 159
ESCNGetDeSkew	p. 163	ESCNGetDocumentSize	p. 165
ESCNGetMaxCropAreas	p. 168	ESCNGetRemainingImages	p. 172
ESCNGetRotate	p. 161	ESCNRetrievImage	p. 171
ESCNSetAutoSize	p. 158	ESCNSetCutSize	p. 160
ESCNSetDeSkew	p. 164	ESCNSetDocumentSize	p. 166
ESCNSetRotate	p. 162	ESCNSoreImage	p. 169
<b>G</b>			
GetBarcodeData	p. 107	GetCounter	p. 182
GetIQAResult	p. 154	GetMicrText	p. 104
GetOcrABText	p. 105	GetOfflineCodeByIndex	p. 189
GetPrintAlignment	p. 128	GetPrintControl	p. 118
GetPrintCutSheetSettings	p. 138	GetPrintImageMethod	p. 121
GetPrintPosition	p. 113	GetPrintSize	p. 117
GetPrintStation	p. 111	GetPrnCapability	p. 87
GetRealStatus	p. 65	GetScanImage	p. 106
GetTransactionNumber	p. 150	GetType	p. 188
GetVersion	p. 155		
<b>I</b>			
InkHeadCleaning	p. 179	InsertValidation	p. 136
<b>L</b>			
LoadAPISettings	p. 184	LoadTemplatePrintArea	p. 144
<b>M</b>			
MICRCleaning	p. 178	MICRClearSpaces	p. 102
MICRGetStatus	p. 177	MICRSelectDataHandling	p. 67
<b>O</b>			
OpenDrawer	p. 180	OpenMonPrinter	p. 61

API	Page	API	Page
<b>P</b>			
PrintBarCode	<a href="#">p. 123</a>	PrintCutSheet	<a href="#">p. 140</a>
PrintImage	<a href="#">p. 127</a>	PrintMemoryImage	<a href="#">p. 130</a>
PrintMultipleToneImage	<a href="#">p. 134</a>	PrintText	<a href="#">p. 132</a>
<b>R</b>			
RemoveValidation	<a href="#">p. 137</a>	ResetCounter	<a href="#">p. 183</a>
ResetPrinter	<a href="#">p. 186</a>	RingBuzzer	<a href="#">p. 181</a>
<b>S</b>			
SCNDeleteCroppingArea	<a href="#">p. 85</a>	SCNGetClampStatus	<a href="#">p. 190</a>
SCNGetCroppingArea	<a href="#">p. 96</a>	SCNGetImageAdjustment	<a href="#">p. 98</a>
SCNGetImageFormat	<a href="#">p. 88</a>	SCNGetImageQuality	<a href="#">p. 92</a>
SCNGetImageTypeOption	<a href="#">p. 73</a>	SCNGetScanArea	<a href="#">p. 90</a>
SCNMICRCancelFunction	<a href="#">p. 84</a>	SCNMICRCancelStatusBack	<a href="#">p. 63</a>
SCNMICRFunction	<a href="#">p. 80</a>	SCNMICRFunctionContinuously	<a href="#">p. 74</a>
SCNMICRFunctionPostPrint	<a href="#">p. 79</a>	SCNPrintMemoryImage	<a href="#">p. 131</a>
SCNPrintText	<a href="#">p. 133</a>	SCNSelectScanFace	<a href="#">p. 86</a>
SCNSelectScanImage	<a href="#">p. 95</a>	SCNSelectScanUnit	<a href="#">p. 69</a>
SCNSetCroppingArea	<a href="#">p. 97</a>	SCNSetImageAdjustment	<a href="#">p. 100</a>
SCNSetImageFormat	<a href="#">p. 89</a>	SCNSetImageQuality	<a href="#">p. 93</a>
SCNSetImageTypeOption	<a href="#">p. 70</a>	SCNSetScanArea	<a href="#">p. 91</a>
SelectErrorEjectAtContinuously	<a href="#">p. 175</a>	SelectJamDetect	<a href="#">p. 176</a>
SetBehaviorToScnResult	<a href="#">p. 173</a>	SetConfigure	<a href="#">p. 185</a>
SetEndorseDirection	<a href="#">p. 115</a>	SetMonInterval	<a href="#">p. 110</a>
SetNumberOfDocuments	<a href="#">p. 174</a>	SetOcrABAreaOrigin	<a href="#">p. 103</a>
SetPaperThickness	<a href="#">p. 191</a>	SetPrintAlignment	<a href="#">p. 129</a>
SetPrintControl	<a href="#">p. 119</a>	SetPrintCutSheetSettings	<a href="#">p. 139</a>
SetPrintImageMethod	<a href="#">p. 122</a>	SetPrintPosition	<a href="#">p. 114</a>
SetPrintSize	<a href="#">p. 120</a>	SetPrintStation	<a href="#">p. 112</a>
SetTemplatePrintArea	<a href="#">p. 147</a>	SetTransactionNumber	<a href="#">p. 151</a>
SetTransactionNumberWithIncremental	<a href="#">p. 152</a>	SetWaterfallMode	<a href="#">p. 153</a>
StartEndorsementCancelStatusBack	<a href="#">p. 64</a>		
<b>T</b>			
TemplatePrint	<a href="#">p. 145</a>		
<b>U</b>			
UpdateEndorseText	<a href="#">p. 135</a>		

## MFDevice Class -Constructor-

Constructor and instance of the MFDevice class, connect to device if the extended version is used.



If an expansion constructor is used, check the results with [LastError \(p.192\)](#) and [IsValid \(p.192\)](#).

### Syntax

- **MFDevice** ()
- **MFDevice** (OpenType type, String name)

Argument

OpenType type:

This Determine type of value supplied in name.

type	Description
TYPE_PORT	Specify the port name in pName.
TYPE_PRINTER	Specify the device name in pName.

String name:

Port name or device name to open.

- If the nType is TYPE\_PORT, it specifies the port name to which the device is connected.

```
<TM-S9000>
```

```
MFDevice (TYPE_PORT, "USB3");
```

```
<TM-S2000>
```

```
MFDevice (TYPE_PORT, "USB4");
```

- If the nType is TYPE\_PRINTER, the device name is specified.

```
<TM-S9000>
```

```
MFDevice(TYPE_PRINTER, "TM-S9000U");
```

```
<TM-S2000>
```

```
MFDevice(TYPE_PRINTER, "TM-S2000U");
```



If the type is TYPE\_PRINTER, you can automatically detect and control devices connected to the computer. In that case, set name as "TM-S9000U, TM-S2000U".

### Description

If the extended constructor is used methods other than OpenMonPrinter will be available for use.

# ResetLastError

Reset the last recorded error.

---

**Syntax**

```
void ResetLastError()
```

---

**Description**

After this method is called the LastError property will return SUCCESS until an error occurs.

# OpenMonPrinter

This API enables communication with a device connected to local computers.

## Syntax

```
ErrorCode OpenMonPrinter(OpenType type, String name)
```

## Argument

**type:** This specifies the type of name specified in pName. One of the following two types is specified.

Constant	Value	Description
TYPE_PORT	1	Specify the port name in name.
TYPE_PRINTER	2	Specify the device name in name.

**name:** This specifies the device that is opened. The specification is as follows, depending on the nType value.

- If the type is TYPE\_PORT, it specifies the port name to which the device is connected.

<TM-S9000>

```
ErrorCode = OpenMonPrinter(TYPE_PORT, "USB3");
```

<TM-S2000>

```
ErrorCode = OpenMonPrinter(TYPE_PORT, "USB4");
```

- If the type is TYPE\_PRINTER, the device name is specified.

<TM-S9000>

```
ErrorCode = OpenMonPrinter(TYPE_PRINTER, "TM-S9000U");
```

<TM-S2000>

```
ErrorCode = OpenMonPrinter(TYPE_PRINTER, "TM-S2000U");
```



If the type is TYPE\_PRINTER, you can automatically detect and control devices connected to the computer. In that case, set name as "TM-S9000U, TM-S2000U".

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_TYPE	-10	nType parameter error
ERR_OPENED	-20	The specified device is already open
ERR_NO_PRINTER	-30	The specified device cannot be found
ERR_NO_TARGET	-40	An unsupported device was specified (The device's power is not On or the cable connections are faulty, etc.)
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_UNKNOWN	-9999	Invalid install configuration



- For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).
- If the common memory has failed to be secured because of the user authority, ERR\_NO\_PRINTER or ERR\_NO\_MEMORY is returned.

## CloseMonPrinter

Disconnect from the monitored device.

### Syntax

```
ErrorCode CloseMonPrinter( )
```

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SCNMICRCancelStatusBack

Disable the SCNMICRStatusCallback event.

### Syntax

```
ErrorCode SCNMICRCancelStatusBack( )
```

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# StartEndorsementCancelStatusBack

Cancels the processing status information notification request registered using StartEndorsementStatusCallback.

---

## Syntax

ErrorCode **StartEndorsementCancelStatusBack**( )

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# EndEndorsementCancelStatusBack

Cancels the processing status information notification request registered using EndEndorsementStatusCallback.

---

## Syntax

ErrorCode **EndEndorsementCancelStatusBack**( )

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetRealStatus

Acquires the real-time device status. The device status is set to the lpStatus.



Refer to [Device Status \(p.274\)](#), for the device statuses that you can acquire.

## Syntax

```
ErrorCode GetRealStatus(out ASB asb)
```

## Argument

asb: The real-time status of the device is set. Combination of device status bits.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# CancelStatusBack

Disable the StatusCallback and StatusCallbackEx events.

---

## Syntax

ErrorCode **CancelStatusBack** ( )

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value



- For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).
- Returns "SUCCESS" even if you call this API by mistake while the automatic status notification request process has not been registered.

# CancelInkStatusBack

The ink status notification request is deregistered.

---

## Syntax

ErrorCode **CancelInkStatusBack** ( )

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value



- For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).
- Returns "SUCCESS" even if you call this API by mistake while the ink status notification request process has not been registered.

# MICRSelectDataHandling

Selects the check reading operation.



This API is compatible with the TM-J9000/TM-S1000 API.

## Syntax

```
ErrorCode MICRSelectDataHandling(CharSelect charSel,
    DetailSelect detailSel, ErrorSelect errorSel)
```

## Argument

charSel: Specifies handling of characters that cannot be analyzed.

Value	Description
0	Interrupts analysis processing at the point when characters that cannot be analyzed are detected and does not add the reading data.
1	Replaces characters which cannot be analyzed with a '?' and continues analysis processing, then if the reading data size is at or less than the reading data size specified in MICRSetReadBackFunction / MICRSetReadBackWnd, the reading data are added.

detailSelect: Not Used.

errorSel: Specifies whether to finish or to continue the reading process when an error occurs. However, the reading process continues when it finishes normally or when an error that adds reading results occurs, irrespective of this setting. The following values can be set individually or as a logical sum.

Constant	Value	Description
ES_STOP_ALL	0	Finishes the reading process when an error occurs.
ES_CONTINUE_ALL	1	Continues the reading process when an error occurs if continuing is possible. The 4 errors where continuing is possible are double feed, magnetic waveform detection error, unrecognized character detection error, and noise error.
ES_CONTINUE_DOUBLEFEED	2	Continues the reading process when double feed is detected.
ES_CONTINUE_NODATA	4	Continues the reading process when a magnetic waveform detection error occurs.
ES_CONTINUE_BADDATA	8	Continues the reading process when an unrecognized character detection error occurs.
ES_CONTINUE_NOISE	16	Continues the reading process when a noise error occurs.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value specified for the product is incorrect.
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SCNSelectScanUnit

Sets the unit that operates image scanning.

### Syntax

```
ErrorCode SCNSelectScanUnit(ScanUnit scanUnit)
```

### Argument

scanUnit: Specifies the unit to scan. The selectable value is as follows.

Constant	Value	Description
EPS_BI_SCN_UNIT_CHECKPAPER	48	Unit for check reading
EPS_BI_SCN_UNIT_CARD	49	Unit for card scanning

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSetImageTypeOption

The light source to be used for scan execution is selected.

Scan processing is executed using the light source specified with this API.



When scanning, the light source specified with this API is used to retrieve images.

## Syntax

```
ErrorCode SCNSetImageTypeOption(ImageTypeOption  
                                imagetypeoption)
```

## Argument

imagetypeoption: Image options are specified. The following values can be specified independently:

Constant	Value	Used Light Source
EPS_BI_SCN_OPTION_COLOR (Default)	0	RGB color
EPS_BI_SCN_OPTION_IR	1	Infrared
EPS_BI_SCN_OPTION_DROPOUT_RED	2	Red of RGB color
EPS_BI_SCN_OPTION_DROPOUT_GREEN	3	Green of RGB color
EPS_BI_SCN_OPTION_DROPOUT_BLUE	4	Blue of RGB color
EPS_BI_SCN_OPTION_COLOR_IR	8	RGB color and infrared
EPS_BI_SCN_OPTION_GRAYSCALE	9	RGB grayscale
EPS_BI_SCN_OPTION_GRAYSCALE_IR	10	RGB grayscale and infrared
EPS_BI_SCN_OPTION_DROPOUT_RED_IR	11	Red of RGB color and infrared
EPS_BI_SCN_OPTION_DROPOUT_GREEN_IR	12	Green of RGB color and infrared
EPS_BI_SCN_OPTION_DROPOUT_BLUE_IR	13	Blue of RGB color and infrared

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

**Description**

Some light sources cannot be used, depending on the SCNSelectScanUnit settings.

imagetypeoption	SCNSelectScanUnit	
	Check Paper	Card
EPS_BI_SCN_OPTION_COLOR	✓	✓
EPS_BI_SCN_OPTION_IR	✓	✓
EPS_BI_SCN_OPTION_DROPOUT_RED	✓	✓
EPS_BI_SCN_OPTION_DROPOUT_GREEN	✓	✓
EPS_BI_SCN_OPTION_DROPOUT_BLUE	✓	✓
EPS_BI_SCN_OPTION_COLOR_IR	✓	-
EPS_BI_SCN_OPTION_GRAYSCALE	✓	✓
EPS_BI_SCN_OPTION_GRAYSCALE_IR	✓	-
EPS_BI_SCN_OPTION_DROPOUT_RED_IR	✓	-
EPS_BI_SCN_OPTION_DROPOUT_GREEN_IR	✓	-
EPS_BI_SCN_OPTION_DROPOUT_BLUE_IR	✓	-

- Image Data Type sent from firmware to driver

Parameter of this API	Color of light source				# of image plane per side	Byte per pixel in visible light source image
	Red	Green	Blue	Infrared		
EPS_BI_SCN_OPTION_COLOR	On	On	On	-	1	3
EPS_BI_SCN_OPTION_IR	-	-	-	On	1	n/a
EPS_BI_SCN_OPTION_DROPOUT_RED	On	-	-	-	1	1
EPS_BI_SCN_OPTION_DROPOUT_GREEN	-	On	-	-	1	1
EPS_BI_SCN_OPTION_DROPOUT_BLUE	-	-	On	-	1	1
EPS_BI_SCN_OPTION_COLOR_IR	On	On	On	On	2	3
EPS_BI_SCN_OPTION_GRAYSCALE	On	On	On	-	1	1
EPS_BI_SCN_OPTION_GRAYSCALE_IR	On	On	On	On	2	1
EPS_BI_SCN_OPTION_DROPOUT_RED_IR	On	-	-	On	2	1
EPS_BI_SCN_OPTION_DROPOUT_GREEN_IR	-	On	-	On	2	1
EPS_BI_SCN_OPTION_DROPOUT_BLUE_IR	-	-	On	On	2	1

- Image Data Type sent to application

\* inconsistent setting

2nd parameter of SCNSelectScanImage	Parameter of this API	2nd parameter of SCNSetImageQuality	Image Type	Bit per pixel
MF_SCAN_IMAGE_VISIBLE	<ul style="list-style-type: none"> <li>• EPS_BI_SCN_OPTION_COLOR</li> <li>• EPS_BI_SCN_OPTION_COLOR_IR</li> </ul>	EPS_BI_SCN_1BIT	Black & White	1
		EPS_BI_SCN_8BIT	Grayscale	8
		EPS_BI_SCN_24BIT	Color	24
	<ul style="list-style-type: none"> <li>• EPS_BI_SCN_OPTION_GRAYSCALE</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_RED</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_GREEN</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_BLUE</li> <li>• EPS_BI_SCN_OPTION_GRAYSCALE_IR</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_RED_IR</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_GREEN_IR</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_BLUE_IR</li> </ul>	EPS_BI_SCN_1BIT	Black & White	1
		EPS_BI_SCN_8BIT	Grayscale	8
		EPS_BI_SCN_24BIT	Grayscale	8*
		EPS_BI_SCN_24BIT	Grayscale	8*
MF_SCAN_IMAGE_INFRARED	<ul style="list-style-type: none"> <li>• EPS_BI_SCN_OPTION_GRAYSCALE</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_RED</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_GREEN</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_BLUE</li> <li>• EPS_BI_SCN_OPTION_GRAYSCALE_IR</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_RED_IR</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_GREEN_IR</li> <li>• EPS_BI_SCN_OPTION_DROPOUT_BLUE_IR</li> </ul>	EPS_BI_SCN_1BIT	Black & White	1
		EPS_BI_SCN_8BIT	Grayscale	8
		EPS_BI_SCN_24BIT	Grayscale	8*
		EPS_BI_SCN_1BIT	Black & White	1
		EPS_BI_SCN_8BIT	Grayscale	8
		EPS_BI_SCN_24BIT	Grayscale	8*
	<ul style="list-style-type: none"> <li>• EPS_BI_SCN_OPTION_IR</li> <li>• EPS_BI_SCN_OPTION_COLOR_IR</li> </ul>	EPS_BI_SCN_1BIT	Black & White	1
		EPS_BI_SCN_8BIT	Grayscale	8
		EPS_BI_SCN_24BIT	Grayscale	8*
		EPS_BI_SCN_24BIT	Grayscale	8*

# SCNGetImageTypeOption

Acquires the action of the light source during scan that is retained in the driver.



A value specifiable by SCNSetImageTypeOption can be acquired.

## Syntax

```
ErrorCode SCNGetImageTypeOption(out ImageTypeOption
    imagetypeoption)
```

## Argument

imagetypeoption: Specifies the memory address where the action of the light source during scan is set. The set value is as shown below.

Constant	Value	Used Light Source
EPS_BI_SCN_OPTION_COLOR	0	RGB color
EPS_BI_SCN_OPTION_IR	1	Infrared
EPS_BI_SCN_OPTION_DROPOUT_RED	2	Red of RGB color
EPS_BI_SCN_OPTION_DROPOUT_GREEN	3	Green of RGB color
EPS_BI_SCN_OPTION_DROPOUT_BLUE	4	Blue of RGB color
EPS_BI_SCN_OPTION_COLOR_IR	8	RGB color and infrared
EPS_BI_SCN_OPTION_GRAYSCALE	9	RGB grayscale
EPS_BI_SCN_OPTION_GRAYSCALE_IR	10	RGB grayscale and infrared
EPS_BI_SCN_OPTION_DROPOUT_RED_IR	11	Red of RGB color and infrared
EPS_BI_SCN_OPTION_DROPOUT_GREEN_IR	12	Green of RGB color and infrared
EPS_BI_SCN_OPTION_DROPOUT_BLUE_IR	13	Blue of RGB color and infrared

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value specified for the product is incorrect.
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNMICRFunctionContinuously

Scans images successively, and reads MICR characters.



- This API allows for the scanning of ID cards.
- When scanning cards, IQA is not executed even if it has been set.
- The endorsement done with this API is carried out only on the station set immediately before.
- When printing the endorsement done with this API, set the station with SetPrintStation before executing this API.

## Syntax

- ErrorCode **SCNMICRFunctionContinuously**  
(FunctionType functionType)
- ErrorCode **SCNMICRFunctionContinuously**  
(MF mf, FunctionType functionType)

## Argument

functionType: Specifies the functions for the API to execute. For settings regarding reading, by specifying MF\_SET\_MICR\_PARAM and so on and executing this function, the API is notified. If the members of the class are changed after notification, it is necessary to perform notification again with this function.

FunctionType functionType

functionType (Constant)	Description
MF_EXEC	Executes check paper scan.
MF_SET_BASE_PARAM	Sets the MFBase class to the driver.
MF_SET_MICR_PARAM	Sets the MFMicr class to the driver.
MF_SET_SCAN_FRONT_PARAM	Sets the MFScan class for the front to the driver.
MF_SET_SCAN_BACK_PARAM	Sets the MFScan class for the back to the driver.
MF_SET_PRINT_PARAM	Sets the MFPrint class to the driver.
MF_SET_PROCESS_PARAM	Sets the MFProcess class to the driver.
MF_SET_IQA_PARAM	Sets the MFIqa class to the driver.
MF_SET_BARCODE_FRONT_PARAM	Sets the MFBarcode class for the front to the driver.
MF_SET_BARCODE_BACK_PARAM	Sets the MFBarcode class for the back to the driver.
MF_CLEAR_BASE_PARAM	Clears the set MFBase class.
MF_CLEAR_MICR_PARAM	Clears the set MFMicr class.
MF_CLEAR_SCAN_FRONT_PARAM	Clears the set MFScan class for the front.
MF_CLEAR_SCAN_BACK_PARAM	Clears the set MFScan class for the back.
MF_CLEAR_PRINT_PARAM	Clears the set MFPrint class.
MF_CLEAR_PROCESS_PARAM	Clears the set MFProcess class.
MF_CLEAR_IQA_PARAM	Clears the set MFIqa class.
MF_CLEAR_BARCODE_FRONT_PARAM	Clears the set MFBarcode class for the front.
MF_CLEAR_BARCODE_BACK_PARAM	Clears the set MFBarcode class for the back.
MF_GET_BASE_DEFAULT	Acquires the MFBase class default value.
MF_GET_MICR_DEFAULT	Acquires the MFBase class default value.
MF_GET_SCAN_DEFAULT	Acquires the MFScan class default value (Front).

*FunctionType functionType*

<b>functionType (Constant)</b>	<b>Description</b>
MF_GET_SCAN_FRONT_DEFAULT	Acquires the MFScan class default value for the front.
MF_GET_SCAN_BACK_DEFAULT	Acquires the default value for the MFScan class for the back.
MF_GET_PRINT_DEFAULT	Acquires the MFPrint class default value.
MF_GET_PROCESS_DEFAULT	Acquires the MFProcess class default value.
MF_GET_IQA_DEFAULT	Acquires the MFIqa class default value.
MF_GET_BARCODE_FRONT_DEFAULT	Acquires the default value for the MFBarcode class for the front.
MF_GET_BARCODE_BACK_DEFAULT	Acquires the default value for the MFBarcode class for the back.

mf: Sets the each class to the instance.

## Return value

*SCNMICRFunctionContinuously*

<b>Macro Definition (Constant)</b>	<b>Value</b>	<b>Description</b>
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_OFFLINE	-110	Waiting to return from an offline state
ERR_WITHOUT_CB	-130	Callback not registered
ERR_PAPERINSERT_TIMEOUT	-300	Failed to insert paper
ERR_EXEC_FUNCTION	-310	Another API is running
ERR_RESET	-400	Device is being reset
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_SCAN	-450	Failed to read the image data
ERR_LINE_OVERFLOW	-460	Line overflow occurred during transaction printing
ERR_PAPER_PILED	-1010	Paper piling error
ERR_PAPER_JAM	-1020	Paper jam error
ERR_COVER_OPEN	-1030	Cover open error
ERR_MICR_NODATA	-1040	MICR data is not existing
ERR_MICR_BADDATA	-1050	MICR data is not able to recognize
ERR_MICR_NOISE	-1070	Noise error has occurred during MICR reading
ERR_SCN_COMPRESS	-1080	Scan image data compressing error
ERR_PAPER_EXIST	-1090	API can not be execute because there is a paper on the path
ERR_PAPER_INSERT	-1100	Failed to insert paper

*MFBase.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_PAPERINSERT_TIMEOUT	-300	Failed to insert paper
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_SCAN	-450	Failed to read the image data
ERR_LINE_OVERFLOW	-460	Line overflow occurred during transaction printing
ERR_NOT_EXEC	-470	Reading process not executed
ERR_PAPER_PILED	-1010	Paper pilling error
ERR_PAPER_JAM	-1020	Paper jam error
ERR_COVER_OPEN	-1030	Cover open error
ERR_MICR_NODATA	-1040	MICR data is not existing
ERR_MICR_BADDATA	-1050	MICR data is not able to recognize
ERR_MICR_NOISE	-1070	Noise error has occurred during MICR reading
ERR_SCN_COMPRESS	-1080	Scan image data compressing error
ERR_PAPER_EXIST	-1090	API can not be execute because there is a paper on the path
ERR_PAPER_INSERT	-1100	Failed to insert paper

*MFScan.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_PARAM	-90	Parameter error
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_SCAN	-450	Device failed in image scanning
ERR_NOT_EXEC	-470	Process not being executed
ERR_SCN_COMPRESS	-1080	Scan image data compressing error

*MFMicr.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_BUFFER_OVER_FLOW	-140	Insufficient buffer error
ERR_NOT_FOUND	-220	Data not found
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_NOT_EXEC	-470	Reading process not executed
ERR_MICR_NODATA	-1040	MICR data is not existing
ERR_MICR_BADDATA	-1050	MICR data is not able to recognize
ERR_MICR_NOISE	-1070	Noise error has occurred during MICR reading

*MFBarcode.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	Data not found
ERR_NOT_EXEC	-470	Process not being executed
ERR_BARCODE_NODATA	-1130	Barcode cannot be detected

*MFPrint.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_LINE_OVERFLOW	-460	Line overflow occurred during transaction printing
ERR_NOT_EXEC	-470	Reading process not executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

### Processing status list

Main status	Sub status	Outline
MF_FUNCTION_START	-	The check sheet read processing was started.
MF_CHECKPAPER_PROCESS_START	-	The check sheet was inserted and the processing was started.
MF_DATARECEIVE_START	-	The data reception processing was started.
MF_DATARECEIVE_DONE	-	The data reception processing was completed.
MF_CHECKPAPER_PROCESS_DONE	-	The check sheet was ejected and the processing was completed.
MF_FUNCTION_DONE	MFBASE.Ret	The check sheet read processing was completed.
MF_ERROR_OCCURED	ERR_PAPER_PILED	Double feed was detected. (Load results: bit5)
	ERR_FORM_LENGTH	A paper length error occurred. (Detailed load results: 44H, 45H)
	ERR_PAPER_JAM	A paper jam error occurred. (Detailed load results: 46H)
	ERR_MECHANICAL	A mechanical error occurred. (Detailed load results: 47H)
	ERR_COVER_OPEN	The processing was terminated because the cover was opened. (Detailed load results: 48H)
	ERR_MICR_NODATA	Magnetic waveform detection error (MICR details: 45H)
	ERR_MICR_BADDATA	Unanalyzable character detection error (MICR details: 46H)
	ERR_MICR_NOISE	Noise error (MICR details: 47H)
	ERR_PRINT_DATA_LENGTH_EXCEED	the printing content exceeds printable area
	ERR_PRINT_DATA_UNRECEIVED	Print data unreceived error occurred.
	ERR_SCN_COMPRESS	Data compression error (SCN details: 47H)
	ERR_SCN_IQA	NOT_PASS is detected at the IQA validation.
ERR_BARCODE_NODATA	Barcode cannot be detected. (BARCODE details: 45H)	

## SCNMICRFunctionPostPrint

Scans images, and reads MICR characters.



- This API allows for the scanning of ID cards.
- When scanning cards, IQA is not executed even if it has been set.
- The endorsement done with this API is carried out only on the station set immediately before.
- When printing the endorsement done with this API, set the station with SetPrintStation before executing this API.

### Syntax

- ErrorCode **SCNMICRFunctionPostPrint**  
(FunctionType functionType)
- ErrorCode **SCNMICRFunctionPostPrint**  
(MF mf, FunctionType functionType)

### Argument

functionType: Specifies the functions for the API to execute. For settings regarding reading, by specifying MF\_SET\_MICR\_PARAM and so on and executing this function, the API is notified. If the members of the class are changed after notification, it is necessary to perform notification again with this function.

Refer to [FunctionType functionType \(p.74\)](#).

mf: Sets the each class to the instance.

### Return value

- ❑ SCNMICRFunctionPostPrint ([p. 75](#))
- ❑ MFBase.Ret ([p. 76](#))
- ❑ MFScan.Ret ([p. 76](#))
- ❑ MFMicr.Ret ([p. 77](#))
- ❑ MFBarcode.Ret ([p. 77](#))
- ❑ MFPrint.Ret ([p. 77](#))



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

### Processing status list

Refer to [Processing status list \(p.78\)](#).

# SCNMICRFunction

Scans images, and reads MICR characters.



- This API is compatible with the TM-J9000/TM-S1000 API.
- This API allows for the scanning of ID cards.

## Syntax

- ErrorCode **SCNMICRFunction** (FunctionType functionType)
- ErrorCode **SCNMICRFunction**  
(MF mf, FunctionType functionType)

## Argument

- mf: Sets the each class to the instance.
- functionType: Specifies the functions for the API to execute. For settings regarding reading, by specifying MF\_SET\_MICR\_PARAM and so on and executing this function, the API is notified. If the members of the class are changed after notification, it is necessary to perform notification again with this function.

### FunctionType functionType

functionType (Constant)	Description
MF_EXEC	Executes check paper scan.
MF_CONTINUE	Executes the scan action to take when double feed is detected.
MF_MICR_RETRANS	Re-acquires the previously scanned MICR data.
MF_SCAN_FRONT_RETRANS	Re-acquires the previously scanned front image.
MF_SCAN_BACK_RETRANS	Re-acquires the previously scanned back image.
MF_SET_BASE_PARAM	Sets the MFBase class to the driver.
MF_SET_MICR_PARAM	Sets the MFMicr class to the driver.
MF_SET_SCAN_FRONT_PARAM	Sets the MFScan class for the front to the driver.
MF_SET_SCAN_BACK_PARAM	Sets the MFScan class for the back to the driver.
MF_SET_PRINT_PARAM	Sets the MFPrint class to the driver.
MF_SET_BARCODE_FRONT_PARAM	Sets the MFBarcode class for the front to the driver.
MF_SET_BARCODE_BACK_PARAM	Sets the MFBarcode class for the back to the driver.
MF_CLEAR_BASE_PARAM	Clears the set MFBase class.
MF_CLEAR_MICR_PARAM	Clears the set MFMicr class.
MF_CLEAR_SCAN_FRONT_PARAM	Clears the set MFScan class for the front.
MF_CLEAR_SCAN_BACK_PARAM	Clears the set MFScan class for the back.
MF_CLEAR_PRINT_PARAM	Clears the set MFPrint class.
MF_CLEAR_BARCODE_FRONT_PARAM	Clears the set MFBarcode class for the front.
MF_CLEAR_BARCODE_BACK_PARAM	Clears the set MFBarcode class for the back.
MF_GET_BASE_DEFAULT	Acquires the MFBase class default value.
MF_GET_MICR_DEFAULT	Acquires the MFBase class default value.
MF_GET_SCAN_DEFAULT	Acquires the MFScan class default value (Front).
MF_GET_SCAN_FRONT_DEFAULT	Acquires the MFScan class default value for the front.

*FunctionType functionType*

<b>functionType (Constant)</b>	<b>Description</b>
MF_GET_SCAN_BACK_DEFAULT	Acquires the default value for the MFScan class for the back.
MF_GET_PRINT_DEFAULT	Acquires the MFPrint class default value.
MF_GET_BARCODE_FRONT_DEFAULT	Acquires the default value for the MFBarcode class for the front.
MF_GET_BARCODE_BACK_DEFAULT	Acquires the default value for the MFBarcode class for the back.

## Return value

*SCNMICRFunction*

<b>Macro Definition (Constant)</b>	<b>Value</b>	<b>Description</b>
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_OFFLINE	-110	Waiting to return from an offline state
ERR_PAPERINSERT_TIMEOUT	-300	Failed to insert paper
ERR_EXEC_FUNCTION	-310	Another API is running
ERR_RESET	-400	Device is being reset
ERR_THREAD		
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_SCAN	-450	Failed to read the image data
ERR_LINE_OVERFLOW	-460	Line overflow occurred during transaction printing
ERR_PAPER_JAM	-1020	Paper jam error
ERR_PAPER_EXIST	-1090	API can not be execute because there is a paper on the path
ERR_PAPER_INSERT	-1100	Failed to insert paper

*MFBBase.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_PAPERINSERT_TIMEOUT	-300	Failed to insert paper
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_SCAN	-450	Failed to read the image data
ERR_NOT_EXEC	-470	Reading process not executed
ERR_PAPER_JAM	-1020	Paper jam error
ERR_PAPER_INSERT	-1100	Failed to insert paper

*MFScan.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_PARAM	-90	Parameter error
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_SCAN	-450	Device failed in image scanning
ERR_NOT_EXEC	-470	Process not being executed

*MFMicr.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_BUFFER_OVER_FLOW	-140	Insufficient buffer error
ERR_NOT_FOUND	-220	Data not found
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_NOT_EXEC	-470	Reading process not executed

*MFBarcode.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	Data not found
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_NOT_EXEC	-470	Process not being executed
ERR_BARCODE_NODATA	-1130	Barcode cannot be detected

*MFPrint.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_NOT_EXEC	-470	Reading process not executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNMICRCancelFunction

SCNMICRFunctionContinuously/SCNMICRFunctionPostPrint is interrupted.

## Syntax

ErrorCode **SCNMICRCancelFunction**(MfEjectType ejectType)

## Argument

ejectType: This parameter is ignored.

wEjectType (Constant)	Parameter to Set for lpvStruct
MF_EJECT_DISCHARGE	Ejects the paper into a pocket.
MF_EJECT_RELEASE	Specifies that the paper should be unclamped and left in the same location.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_OFFLINE	-110	Waiting to return from an offline state
ERR_WITHOUT_CB	-130	Callback not registered
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## Description

- ❑ SCNMICRFunctionContinuously/SCNMICRFunctionPostPrint is interrupted. Alternatively, paper-eject is executed after the ERR\_LINE\_OVERFLOW error occurs with the SCNMICRFunctionContinuously/SCNMICRFunctionPostPrint.
- ❑ When interrupting the SCNMICRFunctionContinuously/SCNMICRFunctionPostPrint, if the interruption is not complete after a maximum of 20 seconds has elapsed, ERR\_TIMEOUT is returned. Once interrupted, the results are set in the storage area for the structure specified by SCNMICRFunctionContinuously/SCNMICRFunctionPostPrint.
- ❑ The only times that ERR\_EXEC\_FUNCTION is returned by this API is when there are overlapping calls of this API with multiple threads and when the device is initialized after print setup and turning the device power off and on.



Paper is ejected into the specified pocket according to the scan settings set for each structure.

## SCNDeleteCroppingArea

Deletes a registered cropping area.

### Syntax

ErrorCode **SCNDeleteCroppingArea**(byte areaNo)

### Argument

areaNo: Specifies the number (0 to 255) of the cropping area to be deleted.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSelectScanFace

Specifies the back or front for an image that is being read.



Scan-type API execute scans using the light source set with this API.

## Syntax

```
ErrorCode SCNSelectScanFace(ScanSide scanFace)
```

## Argument

scanFace: This specifies the back or front for an image that is being read.

Constant	Value	Description
MF_SCAN_FACE_FRONT	0	Specified for the front.
MF_SCAN_FACE_BACK	1	Specified for the back.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).



# SCNGetImageFormat

Acquires the format of the image.

---

## Syntax

ErrorCode **SCNGetImageFormat**(out Format format)

## Argument

format: Specifies the memory address where the format of the notified image data is set. The set value is as shown below.

Constant	Value	Description
EPS_BI_SCN_TIFF	1	TIFF format CCITT (Group 4) compressed data
EPS_BI_SCN_RASTER	2	Raster format uncompressed data
EPS_BI_SCN_BITMAP	3	Bitmap format uncompressed data
EPS_BI_SCN_TIFF256	4	TIFF format uncompressed data
EPS_BI_SCN_JPEGHIGH	5	JPEG format high compression (size priority) data
EPS_BI_SCN_JPEGNORMAL	6	JPEG format normal compression data
EPS_BI_SCN_JPEGLOW	7	JPEG format low compression (quality priority) data
EPS_BI_SCN_JTIFF	8	TIFF format JPEG compressed data

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSetImageFormat

Selects the format of the scanning image data.



The selected format is enabled until CloseMonPrinter is executed.

## Syntax

ErrorCode **SCNSetImageFormat**(Format format)

## Argument

format: Specifies the format of image data notified. The valid specification values are as shown below. The default value is EPS\_BI\_SCN\_TIFF(1).

Constant	Value	Description
EPS_BI_SCN_TIFF	1	TIFF format CCITT (Group 4) compressed data
EPS_BI_SCN_RASTER	2	Raster format uncompressed data
EPS_BI_SCN_BITMAP	3	Bitmap format uncompressed data
EPS_BI_SCN_TIFF256	4	TIFF format uncompressed data
EPS_BI_SCN_JPEGHIGH	5	JPEG format high compression (size priority) data
EPS_BI_SCN_JPEGNORMAL	6	JPEG format normal compression data
EPS_BI_SCN_JPEGLOW	7	JPEG format low compression (quality priority) data
EPS_BI_SCN_JTIFF	8	TIFF format JPEG compressed data



To save as an image file of ANSI X9.100-181-2007 standard, it must be a TIFF format with CCITT(Group 4) compression data (bFormat = EPS\_BI\_SCN\_TIFF), black and white (bColorDepth of SCNSetImageQuality = EPS\_BI\_SCN\_1BIT), and whose resolution is 200 dpi (sResolution of MF\_SCAN = MF\_SCAN\_DPI\_200).

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNGetScanArea

The read area of the device scanner image is acquired.



This API is compatible with the TM-J9000 API.

## Syntax

- ErrorCode **SCNGetScanArea**(out Rectangle area)

### Argument

area: System.Drawing.Rectangle object showing scan area.

- ErrorCode **SCNGetScanArea**(out int startX, out int startY, out int endX, out int endY)

### Argument

startX: Returns the start X coordinate of the read area.

startY: Returns the start Y coordinate of the read area.

endX: Returns the end X coordinate of the read area.

endY: Returns the end Y coordinate of the read area.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSetScanArea

Configures the reading area of the image data.



This API is compatible with the TM-J9000 API.

## Syntax

- ErrorCode **SCNSetScanArea**(Rectangle area)

Argument

area: System.Drawing.Rectangle object showing scan area.

- ErrorCode **SCNSetScanArea**  
(int startX, int startY, int endX, int endY)

Argument

startX: Specifies the start X coordinate (0 to 108) of the read area in units of mm. The initial value is 0.

startY: Specifies the start Y coordinate (0 to 252) of the read area in units of mm. The initial value is 0.

endX: Specifies the end X coordinate (0 to 110, and a value larger than startX) of the read area in units of mm. The initial value is 0. When 0 is specified, the maximum value supported by the device is specified.

endY: Specifies the end Y coordinate (0 to 254, and a value larger than startY) of the read area in units of mm. The initial value is 0. When 0 is specified, the internal measured value of the device is specified.

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNGetImageQuality

Sets the image scanning quality.

## Syntax

```
ErrorCode SCNGetImageQuality  
    (out ColorDepth colorDepth, out double threshold,  
    out Color color, out ExOption exOption)
```

## Argument

- colorDepth: Buffer to store the tonal gradation.
- threshold: Brightness threshold value (-128 ~ 127) for Black and White. Used when colorDepth=EPS\_BI\_SCN\_1BIT, and exOption=EPS\_BI\_SCN\_MANUAL. The value -128 to 127 corresponds to 0 to 255 of the brightness. When "0" is specified, the intermediate brightness "128" is applied.
- color: Buffer to store the color.
- exOption: Specifies density adjustment types. Valid specification values are as shown below.

Constant	Value	Description
EPS_BI_SCN_MANUAL	49	Applies the value of bThreshold for Black and White.
EPS_BI_SCN_SHARP	50	Sharpening
EPS_BI_SCN_SHARP_CUSTOM	51	These constants are compatible with the TM-S1000 API, and as such, do not function with the TM-S9000 and TM-S2000.
EPS_BI_SCN_SHARP_CUSTOM2	52	
EPS_BI_SCN_SHARP_CUSTOM3	53	

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSetImageQuality

Sets the image scanning quality.



If the card unit is selected, when acquiring an image, EPS\_BI\_SCN\_8BIT is acquired, even if EPS\_BI\_SCN\_1BIT is selected.

## Syntax

ErrorCode **SCNSetImageQuality**

(ColorDepth colorDepth, double threshold,  
Color color, ExOption exOption)

## Argument

colorDepth: Specifies the tonal gradation (the number of bits used for 1 pixel).

Constant	Value	Description
EPS_BI_SCN_1BIT	1	Black and White (1 bit)
EPS_BI_SCN_8BIT (Default)	8	256 grayscale (8 bit)
EPS_BI_SCN_24BIT	24	Color (24 bit)

threshold: Brightness threshold value (-128 ~ 127) for Black and White. Used when colorDepth=EPS\_BI\_SCN\_1BIT, and exOption=EPS\_BI\_SCN\_MANUAL. The value -128 to 127 corresponds to 0 to 255 of the brightness. When "0" is specified, the intermediate brightness "128" is applied.

color: Specifies the color. Valid specification values are as shown below. If colorDepth is set to EPS\_BI\_SCN\_1BIT/EPS\_BI\_SCN\_8BIT, specify EPS\_BI\_SCN\_MONOCHROME. If colorDepth is set to EPS\_BI\_SCN\_24BIT, specify EPS\_BI\_SCN\_COLOR.

Constant	Value	Description
EPS_BI_SCN_MONOCHROME	1	Monochrome
EPS_BI_SCN_COLOR	8	Color

exOption: Specifies density adjustment types. Valid specification values are as shown below.

Constant	Value	Description
EPS_BI_SCN_MANUAL	49	Applies the value of bThreshold for Black and White.
EPS_BI_SCN_SHARP	50	Sharpening
EPS_BI_SCN_SHARP_CUSTOM	51	These constants are compatible with the TM-S1000 API, and as such, do not function with the TM-S9000 and TM-S2000. Use EPS_BI_SCN_MANUAL or EPS_BI_SCN_SHARP.
EPS_BI_SCN_SHARP_CUSTOM2	52	
EPS_BI_SCN_SHARP_CUSTOM3	53	

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SCNSelectScanImage

Selects the light source of the scanned image to be retrieved.

The result of a scan executed using the light source specified with this API is returned.



The selected setting is retained as the information on the scan face set with SelectScanFace. If SCNSelectScanFace has been called first, the scan face setting specified with that API is updated.

### Syntax

```
ErrorCode SCNSelectScanImage(ImageType imagetype)
```

### Argument

imagetype: Type of image to be retrieved. The following values can be specified independently:

Constant	Value	Description
MF_SCAN_IMAGE_VISIBLE	0	RGB (visible light) image
MF_SCAN_IMAGE_INFRARED	1	Infrared image

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNGetCroppingAreas

Acquires cropping area information from the device.



This API is compatible with the TM-J9000 API.

## Syntax

- ErrorCode **SCNGetCroppingAreas**(out byte[] areas)

### Argument

areas: Array of bytes with 5 bytes allocated for each cropping area, the first byte is the cropping area index, and the next 4 bytes are the startX, startY, endX and endY values as defined in the SCNSetCroppingArea method.

- ErrorCode **SCNGetCroppingAreas**(out Hashtable areas)

### Argument

areas: Hashtable of Rectangle objects indexed with the cropping area identifier.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_BUFFER_OVER_FLOW	-140	Buffer overflow error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSetCroppingArea

Sets the cropping area.

## Syntax

- ErrorCode **SCNSetCroppingArea**  
(byte areaNo, int startX, int startY,  
int endX, int endY)

### Argument

- areaNo: Specifies the cropping area number (1 to 255).
- startX: Specifies the start X coordinate (0 to 108) of the cropping area in units of mm.
- startY: Specifies the start Y coordinate (0 to 254) of the cropping area in units of mm.
- endX: Specifies the end X coordinate (2 to 110 and larger than startY) of the cropping area in units of mm. When a value that exceeds the area that can be cropped, the maximum value of the croppable area is set.
- endY: Specifies the end Y coordinate (2 to 254 and larger than startY) of the read area in units of mm. When a value that exceeds the area that can be cropped, the maximum value of the croppable area is set.

- ErrorCode **SCNSetCroppingArea**  
(byte areaNo, Rectangle area)

### Argument

- areaNo: Specifies the cropping area number (1 to 255).
- areas: System.Drawing.Rectangle object showing scan area.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNGetImageAdjustment

Acquires the brightness adjustment value, contrast adjustment value, and gamma correction value that are necessary for image correction.

---

## Syntax

```
ErrorCode SCNGetImageAdjustment  
    (out Brightness brightness, out Contrast  
    contrast, out Gamma gamma)
```

## Argument

brightness: Specifies the memory address where the brightness adjustment value is set. The set value is as shown below.

Value	Description
Integer in the range from -100 to 100	-
EPS_BI_SCN_BRIGHTNESS_DEFAULT	Brightness adjustment default value. The setting is the same as the setting when no brightness adjustment is performed.
EPS_BI_SCN_BRIGHTNESS_NONE	Does not perform brightness adjustment.

contrast: Specifies the memory address where the contrast adjustment value is set. The set value is as shown below.

Value	Description
Integer in the range from -100 to 100	-
EPS_BI_SCN_CONTRAST_DEFAULT	Contrast adjustment default value. The setting is the same as the setting when no contrast adjustment is performed.
EPS_BI_SCN_CONTRAST_NONE	Does not perform contrast adjustment.

gamma: Specifies the memory address where the gamma correction value is set. The set value is as shown below.

Value	Description
EPS_BI_SCN_GAMMA_NONE	Does not perform gamma correction.
EPS_BI_SCN_GAMMA_DEFAULT	Gamma correction default value. The setting is the same as the EPS_BI_SCN_GAMMA_NONE.
EPS_BI_SCN_GAMMA_10	Does not perform gamma correction.
EPS_BI_SCN_GAMMA_18	Sets the gamma correction value to 1.8.
EPS_BI_SCN_GAMMA_22	Sets the gamma correction value to 2.2.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNSetImageAdjustment

Sets the brightness adjustment value, contrast adjustment value, and gamma correction value that are needed for image correction.



This API can be called inside the MF\_DATARECEIVE\_DONE callback that is issued when a scan is activated.

## Syntax

ErrorCode **SCNSetImageAdjustment**  
(Brightness brightness, Contrast contrast, Gamma gamma)

## Argument

brightness: Specifies the brightness adjustment value. When 0 is set for brightness, brightness will not be applied.

Value	Description
Integer in the range from -100 to 100	If a value other than 0 is specified, brightness is applied at that value. If 0 is specified, brightness is not applied.
EPS_BI_SCN_BRIGHTNESS_DEFAULT	Brightness adjustment default value. The setting is the same as the setting when no brightness adjustment is performed.
EPS_BI_SCN_BRIGHTNESS_NONE	Does not perform brightness adjustment.

contrast: Specifies the contrast adjustment value. When 0 is set for contrast, contrast will not be applied.

Value	Description
Integer in the range from -100 to 100	If a value other than 0 is specified, contrast is applied at that value. If 0 is specified, contrast is not applied.
EPS_BI_SCN_CONTRAST_DEFAULT	Contrast adjustment default value. The setting is the same as the setting when no contrast adjustment is performed.
EPS_BI_SCN_CONTRAST_NONE	Does not perform contrast adjustment.

gamma: Specifies the Gamma correction value. When 0 is set for gamma, gamma will not be applied.

Value	Description
EPS_BI_SCN_GAMMA_NONE	Does not perform gamma correction.
EPS_BI_SCN_GAMMA_DEFAULT	Gamma correction default value. The setting is the same as the EPS_BI_SCN_GAMMA_NONE.
EPS_BI_SCN_GAMMA_10	Does not perform gamma correction.
EPS_BI_SCN_GAMMA_18	Sets the gamma correction value to 1.8.
EPS_BI_SCN_GAMMA_22	Sets the gamma correction value to 2.2.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# MICRClearSpaces

Clears spaces included in MICR data obtained using [GetMicrText \(p.104\)](#).

---

## Syntax

ErrorCode **MICRClearSpaces**(RemoveSpace removeSpace)

## Argument

removeSpace: Specifies clearance of spaces in MICR data. Specify the following constants.

Constant	Description
CLEAR_SPACE_DISABLE (Default)	Does not clear spaces
CLEAR_SPACE_ENABLE	Clears spaces

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SetOcrABAreaOrigin

Specifies the origin of the OCR area for the MFOcrAb class.

### Syntax

```
ErrorCode SetOcrABAreaOrigin(OcrOrigin eOcrOrigin)
```

### Argument

eOcrOrigin : Specifies the origin of the OCR area. Specify the following values. The default value is OCR\_ORIGIN\_TOP\_LEFT.

Constant	Description
OCR_ORIGIN_TOP_LEFT	Sets the origin to the top left corner in relation to the document insertion direction.
OCR_ORIGIN_BOTTOM_LEFT	Sets the origin to the bottom left corner in relation to the document insertion point.
OCR_ORIGIN_TOP_RIGHT	Sets the origin to the top right corner in relation to the document insertion point.
OCR_ORIGIN_BOTTOM_RIGHT	Sets the origin to the bottom right corner in relation to the document insertion point.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetMicrText

Gets MICR text or OCR text.

---

## Syntax

```
ErrorCode GetMicrText  
    (int transactionNumber, MFMicr mfMicr)
```

## Argument

transactionNumber: Specifies the transaction number (ID) for the MICR text acquired. This is a DWORD type.

mfMicr: A buffer to store the MICR string.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_BUFFER_OVERFLOW	-140	Buffer overflow error
ERR_NOT_FOUND	-220	No data error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_NOT_EXEC	-470	Process not being executed
ERR_MICR_NODATA	-1040	MICR data is not existing
ERR_MICR_BADDATA	-1050	MICR data is not able to recognize



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetOcrABText

Performs the OCR recognition for the OCR-A font or the OCR-B font and acquires the result.



This API only supports the images that is created by the driver. Do not pass the edited image.

## Syntax

```

ErrorCode GetOcrABText
    (uint dwTransactionNumber,
    OcrImageSource eImageSource, String
    szFileName, MFOcrAB mfOcrAB)
  
```

## Argument

**dwTransactionNumber:**

Specifies the transaction number (ID) for the scan image acquired.

**eImageSource:**

Specify an image targeted for the OCR recognition. One of the following values can be specified.

Constant	Description
OCR_SOURCE_TRANSACTION_NUMBER	For an image file stored in the driver.
OCR_SOURCE_IMAGE_FILE	For an image file saved by the driver.

**szFileName:** Specify an image file name targeted for the OCR recognition.

**mfOcrAB:** Buffer to store the information OCR-AB.  
[MFOcrAb Class - Properties \(p.263\)](#).

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	No data error
ERR_IMAGE_FILEOPEN	-230	Open failure
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_MICR_NODATA MICR	-1040	data is not existing



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetScanImage

Gets the scan image.

---

## Syntax

```
ErrorCode GetScanImage  
    (int transactionNumber, MFScan mfScan)
```

## Argument

transactionNumber:

Specifies the transaction number (ID) for the scan image acquired. This is a DWORD type.

mfScan:

A buffer to store the scan results.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	No data error
ERR_SCAN	-450	Device failed in image scanning
ERR_NOT_EXEC	-470	Process not being executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetBarcodeData

Decodes a barcode from the scanned image data, and obtains the result.



The result of barcode decode is set in the MFBarcode class. For the details, refer to [MFBarcode Class - Properties \(p.252\)](#).

## Syntax

```
ErrorCode GetBarcodeData
    (int dwTransactionNumber, MFBarcode mfBarcode)
```

## Argument

dwTransactionNumber:

Specifies a transaction number (ID) for the BARCODE decode.

mfBarcode:

Buffer to store the results of decoding the BARCODE.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	No data error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_BARCODE_NODATA	-1130	Barcode cannot be detected.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# DecodeBarcode

Decodes a barcode from the specified image file, and obtains the result.



The result of barcode decode is set in the MFBarcode class. For the details, refer to [MFBarcode Class - Properties \(p.252\)](#).

---

## Syntax

```
ErrorCode DecodeBarcode  
        (String szFileName, MFBarcode mfBarcode)
```

## Argument

szFileName: Specify an image file name targeted for the barcode decoding.  
mfBarcode: Buffer to store the results of decoding the BARCODE.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_IMAGE_FILEOPEN	-230	Open failure
ERR_IMAGE_UNKNOWNFORMAT	-240	Format injustice
ERR_BARCODE_NODATA	-1130	Barcode cannot be detected.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# DecodeBarcodeMemory

Decodes the barcode from the image data on the application memory, and obtains the result.



The result of barcode decode is set in the MFBarcode class. For the details, refer to [MFBarcode Class - Properties \(p.252\)](#).

## Syntax

```
ErrorCode DecodeBarcodeMemory
        (Bitmap objBitmap, MFBarcode mfBarcode)
```

## Argument

objBitmap: Buffer to store the results of decoding the barcode.  
mfBarcode: Buffer to store the results of decoding the barcode.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_IMAGE_UNKNOWNFORMAT	-240	Format injustice
ERR_BARCODE_NOTDATA	-1130	Barcode cannot be detected.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetMonInterval

This API is compatible. No operation is possible with the TM-S9000/S2000 API.



This API is compatible with the TM-J9000/TM-S1000 API.

## Syntax

```
ErrorCode SetMonInterval  
(int noPrnInterval, int prnInterval)
```

## Argument

noPrnInterval: Not used  
prnInterval: Spooler status monitoring interval is set in units of msec.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetPrintStation

Gets the set station for printing.

---

## Syntax

ErrorCode **GetPrintStation**(out PrintingStation station)

## Argument

station: A value specifiable by SetPrintStation can be acquired.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPrintStation

Sets the station for printing.

Specifies the PrintText, and PrintImage stations for printing.



- The following settings will be retained for each station that has been set with this API:
  - \* LoadTemplatePrintArea execution results
  - \* SetPrintPosition setting
  - \* SetPrintSize setting
  - \* SetPrintAlignment setting
- SCNMICRFuncitonContinuously/PrintCutSheet cannot change the station with a callback during execution. Before executing each API, please be sure to set the print station with this API.

## Syntax

ErrorCode **SetPrintStation**(PrintingStation station)

## Argument

station: Specifies the station for printing. One of the following values can be set.

Constant	Description
MF_ST_ROLLPAPER (Not supported in TM-S2000)	Sets roll paper as the station for printing.
MF_ST_VALIDATION	Sets validation as the station for printing.
MF_ST_E_ENDORSEMENT	Sets electronic endorsement as the station for printing.
MF_ST_E_ENDORSEMENT_BACK	Sets back face electronic endorsement as the station for printing. (Same as MF_ST_E_ENDORSEMENT.)
MF_ST_E_ENDORSEMENT_FRONT	Sets front face electronic endorsement as the station for printing.
MF_ST_PHYSICAL_ENDORSEMENT	Sets physical endorsement as the station for printing.
MF_ST_FEEDER	Sets U-shaped mechanism as the station for printing.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetPrintPosition

Gets the currently set printing start position.

## Syntax

```
ErrorCode GetPrintPosition
(out int horizontal, out int vertical)
```

## Argument

horizontal: Buffer to store the horizontal printing start position.  
vertical: Buffer to store the vertical printing start position.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPrintPosition

Sets the printing start position.



- If electronic endorsement is selected for SetPrintStation, this API can be used.
- If data is being buffered with TemplatePrint, nothing will happen even when this API is executed.

## Syntax

```
ErrorCode SetPrintPosition  
        (int horizontal, int vertical)
```

## Argument

horizontal: Horizontal printing start position (unit: mm).  
vertical: Vertical printing start position (unit: mm).

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetEndorseDirection

Specifies the direction of electronic endorsement.

## Syntax

```
ErrorCode SetEndorseDirection
    (ElectricEndorseDirection eDirection)
```

## Argument

eDirection: Specifies the direction for electronic endorsement. One of the following values can be set.

Constant	Value	Description
EENDORSE_DIRECTION_LEFTRIGHT	1	From left to right (normal direction)
EENDORSE_DIRECTION_TOPBOTTOM	2	From top to bottom (Rotate 90° clockwise)
EENDORSE_DIRECTION_RIGHTLEFT	3	From right to left (upside down)
EENDORSE_DIRECTION_BOTTOMTOP	4	From bottom to top (Rotate 90° counterclockwise)

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# BufferedPrint

Switches to the buffered print mode and executes buffered printing.



If physical endorsement is selected for SetPrintStation, this API cannot be used.

## Syntax

ErrorCode **BufferedPrint**(PrintBuffer function)

## Argument

function: Specifies the function to execute. One of the following values can be set.

Constant	Description
MF_PRT_BUFFERING	Buffers the print data.
MF_PRT_EXEC	Prints the buffered print data.
MF_PRT_CLEAR	Clear the buffered print data and exits buffering mode.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetPrintSize

Gets the print size set in the station specified in SetPrintStation.

## Syntax

```
ErrorCode GetPrintSize(out int width, out int height)
```

## Argument

width: Buffer to store the horizontal print size.  
height: Buffer to store the vertical print size.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetPrintControl

Gets the currently set print control method.



This API is compatible with the TM-J9000 API.

## Syntax

ErrorCode **GetPrintControl**(out PrintSpeed speed, out PrintDirection direction)

## Argument

speed: Buffer to store the currently set printing speed and print quality.  
direction: Buffer to store the currently set ink-jet head control method.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPrintControl

Sets the print control method.



This API is compatible with the TM-J9000 API.

## Syntax

```
ErrorCode SetPrintControl
(PrintSpeed speed, PrintDirection direction)
```

## Argument

speed : Specifies the printing speed and print quality. One of the following values can be set.

Constant	Description
MF_PRINT_SPEED_NORMAL	Prints with high quality.
MF_PRINT_SPEED_HIGH	Prints with high speed.
MF_PRINT_SPEED_ECONOMY	Prints saving ink.

direction: Specifies the ink-jet head control method. One of the following values can be set.

Constant	Description
MF_PRINT_DIRECTION_DOUBLE	Prints in double direction. There may be some slight overlap between passes (lines), but printing is fast.
MF_PRINT_DIRECTION_SINGLE	Prints in single direction. Printing is slow, but there is no overlap between passes (lines).



The TM-S9000/S2000 API ignores this parameter.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPrintSize

Sets the print size.



If MF\_ST\_PHYSICAL\_ENDORSEMENT is selected for SetPrintStation, use [TemplatePrint](#) (p.145).

## Syntax

```
ErrorCode SetPrintSize(int width, int height)
```

## Argument

width: Horizontal print size (unit: mm).



- If MF\_ST\_ROLLPAPER is set for the station, a whole number from 0 to 72 can be designated.(TM-S9000 only)
- If MF\_ST\_VALIDATION is set for the station, a whole number from 0 to 104 can be designated.
- If MF\_ST\_E\_ENDORSEMENT is set for the station, a whole number from 0 to 255 can be designated.
- If MF\_ST\_FEEDER is set for the station, a whole number from 0 to the width of the possible printing range designated in the API Setting File/10 can be designated.

height: Vertical print size (unit: mm).



- If MF\_ST\_ROLLPAPER is set for the station, a whole number from 0 to 65535 can be designated. If a value which exceeds the possible setting range is set, an error is not returned. The information of the low two bytes is set. Ex:) If 65540 is set, because 0x00010004, the information of low two bytes, 4, is set.(TM-S9000 only)
- If MF\_ST\_VALIDATION is set for the station, a whole number from 0 to 27 can be designated.
- If MF\_ST\_E\_ENDORSEMENT is set for the station, a whole number from 0 to 255 can be designated.
- If MF\_ST\_FEEDER is set for the station, a whole number from 0 to the height of the possible print range designated in the API Setting File/10 can be designated.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_SIZE	-1000	Size excess error



For a list of return values and troubleshooting actions, refer to [Return value](#) (p.43).

# GetPrintImageMethod

Acquires the print method for SetPrintImageMethod.

## Syntax

```
ErrorCode GetPrintImageMethod
(out PrintImageType method)
```

## Argument

method: Specifies the memory address where the image print method is set. The set value is as shown below.

Constant	Description
PRINTIMAGEMETHOD_BW	Mono color print (already available)
PRINTIMAGEMETHOD_MULTITONE	Multiple tone print

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPrintImageMethod

Selected for each print station printing method used by PrintImage.



- This API is for use with the roll paper printing. It can be used when the SetPrintStation setting is MF\_ST\_ROLLPAPER. If executed on the TM-S2000, it won't operate.
- If SetPrintStation has been called first, the station setting specified with that API is updated.
- The settings specified with this API will be applied to the printing result.

## Syntax

ErrorCode **SetPrintImageMethod**(PrintImageType method)

## Argument

method: How to print images. The following values can be specified independently:

Constant	Description
PRINTIMAGEMETHOD_BW	Mono color print (already available)
PRINTIMAGEMETHOD_MULTITONE	Multiple tone print

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# PrintBarcode

Executes barcode printing. Prints barcodes from the station specified in SetPrintStation. This API is valid for roll paper printer in TM-S9000 only. Not supported in TM-S2000.



- Barcodes cannot be printed from electronic endorsement. If this API is executed when MF\_ST\_E\_ENDORSEMENT is specified in SetPrintStation, the ERR\_NOT\_SUPPORT error is returned.
- Since this API does not add a quiet zone, the application should make allowance for a quiet zone.
- Barcode printing cannot be executed for endorsement printing. If this API is executed with MF\_ST\_PHYSICAL\_ENDORSEMENT specified using SetPrintStation, the error ERR\_NOT\_SUPPORT will be returned.
- Barcode printing cannot be executed for cut sheet printing. If this API is executed with MF\_ST\_FEEDER specified using SetPrintStation, the error ERR\_NOT\_SUPPORT will be returned.

## Syntax

```

ErrorCode PrintBarcode(byte[] data, Symbol symbol,
                        TextPosition textPosition)
ErrorCode PrintBarcode(String data, Symbol symbol,
                        TextPosition textPosition)

```

## Argument

data: Buffer to store barcode data.  
data: String containing barcode data to be printed.



When designating composite barcode, begin by designating 2-dimensional symbol data, then designate 1-dimensional symbol data in order. And designate 0x3b as the separation between the 2-dimensional symbol data and the 1-dimensional symbol data.

Ex:)

The following is composite barcode data where the first 5 bytes are 2-dimensional symbol data, and the data following 0x3b is one-dimensional symbol data.

```
0x41,0x41,0x41,0x41,0x41,0x3b,0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39,0x31
```

dwSymbol: Specifies the type of barcode. One of the following values can be set.

Constant	Barcode type
CODE_UPCA	UPC-A
CODE_UPCE	UPC-E
CODE_EAN13	EAN13
CODE_EAN8	EAN8
CODE_CODE39	Code39
CODE_ITF	ITF
CODE_CODABAR	Codabar
CODE_CODE93	Code93
CODE_CODE128	Code128
CODE_GS1_128	GS1-128
CODE_GS1_DATABAR_OMINIDIRECTIONAL	GS1 Databar Omnidirectional
CODE_GS1_DATABAR_TRUNCATED	GS1 Databar Truncated
CODE_GS1_DATABAR_LIMITED	GS1 Databar Limited
CODE_GS1_DATABAR_EXPANDED	GS1 Databar Expanded
CODE_PDF417	PDF417
CODE_QRCODE	QR code
CODE_MAXICODE	Maxicode
CODE_GS1_DATABAR_STACKED	GS1-Databar Stacked
CODE_GS1_DATABAR_STACKED_OMINIDIRECTIONAL	GS1-Databar Stacked Omnidirectional
CODE_GS1_DATABAR_EXPANDED_STACKED	GS1-Databar Expanded Stacked
CODE_COMPOSIT	Output as a composite (See <a href="#">Barcode symbols that can be made into composites (p.126)</a> .)

dwTextPosition:

Specifies the HRI text position. One of the following values can be set.

Constant	Description
HRI_NONE	No HRI text is added.
HRI_ABOVE	HRI text is added at the top of the barcode.
HRI_BELOW	HRI text is added at the bottom of the barcode.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_SIZE	-1000	Size excess error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

**Description**

Data ranges that can be set for barcodes

Constant	Length	Characters that can be specified
CODE_UPCA	11 to 12	0x30 to 0x39
CODE_UPCE	11 to 12	0x30 to 0x39
CODE_EAN13	12 to 13	0x30 to 0x39
CODE_EAN8	7 to 8	0x30 to 0x39
CODE_CODE39	1 to 255	0x30 to 0x39, 0x41 to 0x5a, 0x20, 0x24, 0x25, 0x2b, 0x2d, 0x2e, 0x2f
CODE_ITF	1 to 255 (odd numbers)	0x30 to 0x39
CODE_CODABAR	1 to 255	0x30 to 0x39, 0x41 to 0x44, 0x24, 0x2b, 0x2d, 0x2e, 0x2f, 0x3a
CODE_CODE93	1 to 255	0x00 to 0x7f
CODE_CODE128	2 to 255	0x00 to 0x7f, <a href="#">Special characters (p.125)</a>

Special characters

Special characters	ASCII
SHIFT	{S
CODE A	{A
CODE B	{B
CODE C	{C
FNC1	{1
FNC2	{2
FNC3	{3
FNC4	{4
{	{{

Barcode symbols that can be made into composites

<b>Constant</b>	<b>Barcode type</b>
CODE_UPCA	UPC-A
CODE_UPCE	UPC-E
CODE_EAN13	EAN13
CODE_EAN8	EAN8
CODE_GS1_128	GS1-128
CODE_GS1_DATABAR_OMINIDIRECTIONAL	GS1 Databar Omnidirectional
CODE_GS1_DATABAR_TRUNCATED	GS1 Databar Truncated
CODE_GS1_DATABAR_LIMITED	GS1 Databar Limited
CODE_GS1_DATABAR_EXPANDED	GS1 Databar Expanded
CODE_GS1_DATABAR_STACKED	GS1-Databar Stacked
CODE_GS1_DATABAR_STACKED_OMINIDIRECTIONAL	GS1-Databar Stacked Omnidirectional
CODE_GS1_DATABAR_EXPANDED_STACKED	GS1-Databar Expanded Stacked

# PrintImage

Executes image printing from a file.



If MF\_ST\_PHYSICAL\_ENDORSEMENT is selected for SetPrintStation, use [SCNPrintMemoryImage \(p.131\)](#).

## Syntax

```
ErrorCode PrintImage(String filename)
```

## Argument

filename: Specifies the location of the image file to print with a full path. Image file that can be specified are below.

- BMP format (Uncompressed image data only)
- JPEG format (Baseline DCT, Progressive)
- TIFF format (CCITT Group 3/Group 4 compressed data, uncompressed data only)



The following formats are not supported.

- \* BMP format (Compressed image data)
- \* JPEG format (Lossless compression, hierarchical coding)
- \* TIFF format (Palette color, PackBits/JPEG/LZW/ZIP compression)

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_IMAGE_FILEOPEN	-230	Open failure
ERR_IMAGE_UNKNOW NFORMAT	-240	Format injustice
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_SIZE	-1000	Size excess error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetPrintAlignment

Gets the inline printing position set in the station specified in SetPrintStation.

---

## Syntax

ErrorCode **GetPrintAlignment**(out Alignment alignment)

## Argument

alignment: Buffer to store the inline printing position.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the product is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPrintAlignment

Sets the inline printing position.



Only the following settings are applied to the print station specified for SetPrintStation in the settings for this API.

- \* MF\_ST\_VALIDATION
- \* MF\_ST\_ROLLPAPER
- \* MF\_ST\_FEEDER

## Syntax

ErrorCode **SetPrintAlignment**(Alignment alignment)

## Argument

alignment: Specifies the inline printing position.

Value	Description
Integer higher than 0 (unit: mm)	Print location shifts a specified amount from the left. MF_ST_VALIDATION: 0 to 104 MF_ST_ROLLPAPER: 0 to 72 MF_ST_FEEDER: 0 to 104
MF_PRINT_ALIGNMENT_LEFT	Prints with left alignment.
MF_PRINT_ALIGNMENT_CENTER	Prints with center alignment.
MF_PRINT_ALIGNMENT_RIGHT	Prints with right alignment.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error.
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_SIZE	-1000	Size excess error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# PrintMemoryImage

Executes image printing from memory. Not supported in TM-S2000.



If physical endorsement is selected for SetPrintStation, use [SCNPrintMemoryImage](#) (p.131).

## Syntax

ErrorCode **PrintMemoryImage**(Bitmap objImage)

## Argument

objImage: Specifies image data (objBitmapSets BitmapObject) for printing.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_IMAGE_UNKNOW NFORMAT	-240	Format injustice
ERR_SIZE	-1000	Size excess error



For a list of return values and troubleshooting actions, refer to [Return value](#) (p.43).

# SCNPrintMemoryImage

Executes image printing from memory.



This API is for use with physical endorsement. It can be used when the SetPrintStation setting is MF\_ST\_PHYSICAL\_ENDORSEMENT.

## Syntax

```
ErrorCode SCNPrintMemoryImage(uint transactionNumber
, Bitmap objImage)
```

## Argument

transactionNumber:

Transaction number (ID) corresponding to the check sheet from which the processing status has been issued.

objImage:

Specifies image data (objBitmapSets BitmapObject) for printing.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	Data not found
ERR_IMAGE_UNKNOW NFORMAT	-240	Format injustice
ERR_SIZE	-1000	Size excess error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# PrintText

Executes text printing.



If MF\_ST\_PHYSICAL\_ENDORSEMENT is selected for SetPrintStation, use [SCNPrintText](#) (p.133).

## Syntax

```
ErrorCode PrintText(String text, MFDecorate decorate)
```

## Argument

text: Buffer to store the text for printing.  
decorate: Text decoration information.  
Refer to [MFDeviceFont Class - Properties](#) (p.216)/ [MFTrueType Class - Properties](#) (p.219)/[MFPrint Class - Properties](#) (p.271).

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value](#) (p.43).

# SCNPrintText

Executes text printing.



This API is for use with physical endorsement. It can be used when the SetPrintStation setting is MF\_ST\_PHYSICAL\_ENDORSEMENT.

## Syntax

```
ErrorCode SCNPrintText(uint transactionNumber
                        , String text, MFDecorate decorate)
```

## Argument

transactionNumber:

Transaction number (ID) corresponding to the check sheet from which the processing status has been issued.

text:

Specifies the memory address where the text for printing is saved.

decorate:

Specifies the MFDeviceFon/MFTrueType class where the text decoration information is saved. Refer to [MFDeviceFont Class - Properties \(p.216\)](#)/[MFTrueType Class - Properties \(p.219\)](#).

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	Data not found
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# PrintMultipleToneImage

Multiple-tone images registered in NVRAM are printed. Not supported in TM-S2000.



- The roll sheet must be selected in advance using SetPrintStation.
- If images registered in NVRAM are larger than the width of the roll sheet, printing is terminated without anything being printed.
- The image size set using SetPrintSize will not be applied, and the image will be output according to the registered image size.

## Syntax

```
ErrorCode PrintMultipleToneImage  
          (byte bKeyCode1, byte bKeyCode2)
```

## Argument

- szKeyCode1: Specify the first key code for the multiple tone image to be printed. A value from 0x20 to 0x7E can be specified.
- szKeyCode2: Specify the second key code for the multiple tone image to be printed. A value from 0x20 to 0x7E can be specified.



If the image for the designated key code is not found, the image is not printed.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Paper other than roll papers is set in the print station.
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# UpdateEndorseText

Updates an endorsement character string.



This API is compatible with the TM-J9000/TM-S1000 API.

## Syntax

- **ErrorCode** **UpdateEndorseText**  
(String [] strings, int [] attributes  
, FontStyle [] fonts, FontScale [] fontSizes)

### Argument

- strings: Specifies an endorsement character string. lpString(0) is the first line, lpString(1) is the second line, and lpString(2) is the 3rd line.
- attributes: Specifies an attribute of the endorsement character string. dwAttribute(0) is the first line, dwAttribute(1) is the second line, and dwAttribute(2) is the 3rd line.
- fonts: Specifies a font of the endorsement character string. wFont(0) is the first line, wFont(1) is the second line, and wFont(2) is the 3rd line.
- fontSizes: Specifies a font size of the endorsement character string. wFontSize (0) is the first line, wFontSize (1) is the second line, and wFontSize (2) is the 3rd line.

- **ErrorCode** **UpdateEndorseText** (MFPrint mfPrint)

### Argument

- mfPrint: Sets the MFPrint class object.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_NOT_EXEC	-470	Process not being executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# InsertValidation

Inserts the paper in validation and makes the product ready for printing.



This API is compatible with the TM-J9000 API.

## Syntax

```
ErrorCode InsertValidation(int timeout)
```

## Argument

timeout: Specifies the timeout time until the paper is inserted in the validation slot. A value from 0 to 300 in second units can be specified. When 0 is specified, no timeout time is set, and the product waits until paper is inserted.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Cannot R/W to product
ERR_PARAM	-90	Parameter error
ERR_PAPERINSERT_TIMEOUT	-300	Paper insertion time exceeded.
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## RemoveValidation

Ejects the paper inserted in validation, and performs monitoring until it is removed from the insertion slot.



This API is compatible with the TM-J9000 API.

### Syntax

```
ErrorCode RemoveValidation(int timeout)
```

### Argument

**timeout:** Specifies the timeout time until the paper is inserted in the validation slot. A value from 0 to 300 in second units can be specified. When 0 is specified, no timeout time is set, and the product waits until paper is inserted.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot R/W to product
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetPrintCutSheetSettings

Acquires the action settings for cut paper printing.



A value specifiable by SetPrintCutSheetSettings can be acquired.

## Syntax

```
ErrorCode GetPrintCutSheetSettings  
(out int Number, out ScanEnable Scan)
```

## Argument

Number: Specifies the memory address where the number of cut papers to print is saved.  
Scan: Specifies the memory address where the option for simultaneous scan execution is set. The set value is as shown below.

Constant	Description
BI_PRINTCUTSHEET_SCAN	Scans print data when performing cut sheet printing.
BI_PRINTCUTSHEET_NOSCAN	Does not scan print data when performing cut sheet printing.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SetPrintCutSheetSettings

Sets the operation of cut sheet paper printing.



The detailed scan settings are specified using [PrintCutSheet \(p.140\)](#).

### Syntax

```
ErrorCode SetPrintCutSheetSettings
(byte Number, ScanEnable scan)
```

### Argument

- Number: Number of cut sheet papers to be printed.  
The following value can be specified independently: Integer in the range from 1 to 255
- scan: Whether or not simultaneous scan is executed.  
The following values can be specified independently:

Constant	Description
BI_PRINTCUTSHEET_SCAN	Scans print data when performing cut sheet printing.
BI_PRINTCUTSHEET_NOSCAN	Does not scan print data when performing cut sheet printing.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# PrintCutSheet

Prints a cut sheet and executes simultaneous scanning.

Print data is specified inside the print ready callback, in the same way as when endorsement printing is executed using `SCNMICRFunctionContinuously`.



- This API is used in the same manner as [SCNMICRFunctionContinuously \(p.74\)](#).
- If a cut sheet is to be printed with simultaneous scanning set using `StartPrintCutSheet`, the scanning will be executed according to the setting specified in this API.
- Notification that an error has occurred is in the callback registered in `SCNMICRStatusCallback`. If notification of an error is required, register the callback in `SCNMICRStatusCallback`.

## Syntax

- `ErrorCode PrintCutSheet(FunctionType functionType)`
- `ErrorCode PrintCutSheet(MF mf, FunctionType functionType)`

## Argument

`mf`: Sets the each class to the instance.

`functionType`: Specifies the functions for the API to execute. For settings regarding reading, by specifying `MF_SET_MICR_PARAM` and so on and executing this function, the API is notified. If the members of the class are changed after notification, it is necessary to perform notification again with this function.

### FunctionType functionType

functionType (Constant)	Description
<code>MF_EXEC</code>	If <code>MF_EXEC</code> is set, the printing of a cut sheet is started.
<code>MF_SET_BASE_PARAM</code>	Sets the <code>MFBASE</code> class to the driver.
<code>MF_SET_SCAN_PRINTAREA_PARAM</code>	Sets information on the specified <code>MFScan</code> class as information on the print side.
<code>MF_SET_SCAN_NOPRINTAREA_PARAM</code>	Sets information on the specified <code>MFScan</code> class as information on the non-print side.
<code>MF_SET_PROCESS_PARAM</code>	Sets the <code>MFProcess</code> class to the driver.
<code>MF_CLEAR_BASE_PARAM</code>	Clears the set <code>MFBASE</code> class.
<code>MF_CLEAR_SCAN_PRINTAREA_PARAM</code>	Clears the set <code>MFScan</code> class for the print side.
<code>MF_CLEAR_SCAN_NOPRINTAREA_PARAM</code>	Clears the set <code>MFScan</code> class for the non-print side.
<code>MF_CLEAR_PROCESS_PARAM</code>	Clears the set <code>MFProcess</code> class.
<code>MF_GET_BASE_DEFAULT</code>	Acquires the <code>MFBASE</code> class default value.
<code>MF_GET_SCAN_PRINTAREA_PARAM_DEFAULT</code>	Acquires the <code>MFScan</code> class default value for the print side.
<code>MF_GET_SCAN_NOPRINTAREA_PARAM_DEFAULT</code>	Acquires the default value for the <code>MFScan</code> class for the non-print side.
<code>MF_GET_PROCESS_DEFAULT</code>	Acquires the <code>MFProcess</code> class default value.

## Return value

*PrintCutSheet*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_OFFLINE	-110	Waiting to return from an offline state
ERR_PAPERINSERT_TIMEOUT	-300	Failed to insert paper
ERR_EXEC_FUNCTION	-310	Another API is running
ERR_RESET	-400	Device is being reset
ERR_THREAD		
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_SCAN	-450	Failed to read the image data
ERR_LINE_OVERFLOW	-460	Line overflow occurred during transaction printing
ERR_PAPER_JAM	-1020	Paper jam error
ERR_PAPER_EXIST	-1090	API can not be execute because there is a paper on the path
ERR_PAPER_INSERT	-1100	Failed to insert paper

*MFBBase.Ret*

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	Cannot read/write to the device
ERR_PARAM	-90	Parameter error
ERR_PAPERINSERT_TIMEOUT	-300	Failed to insert paper
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_MICR	-440	Failed to read the MICR data
ERR_SCAN	-450	Failed to read the image data
ERR_NOT_EXEC	-470	Reading process not executed
ERR_PAPER_JAM	-1020	Paper jam error
ERR_PAPER_INSERT	-1100	Failed to insert paper

MFScan.Ret

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_PARAM	-90	Parameter error
ERR_ABORT	-430	Canceled by SCNMICRCancelFunction
ERR_SCAN	-450	Device failed in image scanning
ERR_NOT_EXEC	-470	Process not being executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# AutoCutRollPaper

Executes auto-cutter for the roll paper. Not supported in TM-S2000.



- If this API is called while data is being buffered using BufferedPrint/FormatPrint, auto-cutter will not be executed when it is called, but will be executed when buffered data is printed.
- The roll sheet must be selected in advance using SetPrintStation.

## Syntax

```
ErrorCode AutoCutRollPaper(AutoCutType autocuttype)
```

## Argument

autocuttype: How to execute auto-cutter. The following values can be specified independently:

Constant	Description
AUTOCUT_CUT	Sets roll paper as the station for printing.
AUTOCUT_FEEDANDCUT	Sets validation as the station for printing.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Paper other than roll sheets is set in the print station
ERR_EXEC_FUNCTION	-310	A scan is being executed
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_PAPER_INSERT	-1100	No paper is set in the print station



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# LoadTemplatePrintArea

Loads a template printing setting file.

The area information loaded with this API can be specified with [SetTemplatePrintArea \(p.147\)](#).



- If correct information and invalid information coexist, only the correct information is loaded.
- If this API is executed again, the previously loaded information will be discarded.
- If more than 255 normal data items exist, only 255 data items will be loaded, with the remaining data items being ignored.

## Syntax

```
ErrorCode LoadTemplatePrintArea  
        (String FilePath, out int LoadNum)
```

## Argument

- FilePath: Template printing setting file to be loaded. The full path to the file must be specified. Even when just the file name or the relative path is specified, it will not be regarded as an error as long as the file can be loaded.
- LoadNum: The number of areas loaded is stored. If an error occurs, nothing will be set. If null is specified, nothing happens.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset
ERR_DATA_INVALID	-480	Not have print area information, contains incorrect information on print area.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# TemplatePrint

Buffers data for template printing as well as prints buffered data.



- The execution status of this API is retained for each station that can be specified using SetPrintStation. If SetPrintPosition is executed while data is being buffered, the value will be not applied.
- If an API for printing is executed while data is being buffered, the data will be saved internally instead of being printed.
- If SetPrintSize is executed while data is being buffered, the value will be not applied.

## Syntax

```
ErrorCode TemplatePrint
    (TemplatePrintMode templateprintmode)
```

## Argument

templateprintmode:

Specifies the execute mode. The following values can be specified independently:

Constant	Description
TEMPLATEPRINT_BUFFERING	<p>Starts the buffering mode of template-specified print. During buffering mode, even if a print-related API is executed, printing is not executed and print information is retained.</p> <ul style="list-style-type: none"> <li>• If TEMPLATEPRINT_BUFFERING is set, format printing will be enabled in buffering mode.</li> <li>• If TEMPLATEPRINT_BUFFERING is set during buffering mode, nothing will happen, and SUCCESS will be returned.</li> </ul>
TEMPLATEPRINT_EXEC	<p>Prints the information retained when in the buffering mode and terminates the buffering mode. After termination, clears all the information specified during buffering.</p> <ul style="list-style-type: none"> <li>• If buffered data is present when TEMPLATEPRINT_EXEC is set, the buffered data will be printed, which will subsequently be cleared along with the area information.</li> <li>• If no buffered data is present when TEMPLATEPRINT_EXEC is set, nothing will happen, and SUCCESS will be returned.</li> <li>• If TEMPLATEPRINT_EXEC is set, buffering mode will be cancelled.</li> </ul>
TEMPLATEPRINT_CLEAR	<p>Clears the retained information and terminates the buffering mode without printing. After termination, clears all the information specified during buffering.</p> <ul style="list-style-type: none"> <li>• If TEMPLATEPRINT_CLEAR is set, buffered data will be cleared along with the area information.</li> <li>• If no buffered data is present when TEMPLATEPRINT_CLEAR is set, nothing will happen, and SUCCESS will be returned.</li> <li>• If TEMPLATEPRINT_CLEAR is set, buffering mode will be cancelled.</li> </ul>

Constant	Description
TEMPLATEPRINT_EXEC_NOCLEAR	<p>Prints the information retained when in the buffering mode. After print execution, while retaining the present information, continues the buffering mode.</p> <ul style="list-style-type: none"> <li>• If buffered data is present when TEMPLATEPRINT_EXEC_NOCLEAR is set, the buffered data will be printed (but the information will not be cleared).</li> <li>• If no buffered data is present when TEMPLATEPRINT_EXEC_NOCLEAR is set, nothing will happen, and SUCCESS will be returned.</li> <li>• If TEMPLATEPRINT_EXEC_NOCLEAR is set, buffering mode will be cancelled.</li> </ul>

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SetTemplatePrintArea

Selects print areas from print area information and sets the selected areas.  
Print-type APIs are executed for the areas specified with this API.



- Any portions that lie outside the print area will not be printed.
- To specify area names, print area information must be loaded in advance with LoadTemplatePrintArea.
- TemplatePrint must be called in advance with buffering mode enabled.
- This API can be used only when TemplatePrint is executed in buffering mode.
- Although the area that lies outside the entire print area can be specified, nothing will be printed.

### Syntax

```

ErrorCode SetTemplatePrintArea
        (AreaSelectMode mode, String areaname
        , MFPrintAreaInfo info
        , int transactionNumber)
  
```

### Argument

mode: How to set areas.

Constant	Description
SELECTPRINTAREA_AREANAME	For the area information, sets the data defined by the area name in the loaded setting data.
SELECTPRINTAREA_DIRECT	For the area information, sets the data defined by the specified class.

areaname: Area name.



- The null can be specified if direct specification is selected.
- The value will be ignored if direct specification is selected.

info: Print area information. Refer to [MFPrintAreaInfo Class - Properties \(p.269\)](#).



- The null can be specified if area name specification is selected.
- The value will be ignored if area specification is selected.

transactionNumber:

Specifies the transaction number (ID) to set.



When executing at stations other than the physical endorsement station, please designate 0x40000000. If a value other than 0x40000000 is designated, the operation is definitely done to the physical endorsement station.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	Data not found
ERR_NOT_EXEC	-470	Process not being executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ClearTemplatePrintData

Clears the print data located in the print area specified with SetTemplatePrintArea.



- TemplatePrint must be called in advance with buffering mode enabled.
- This API can be used only when TemplatePrint is executed in buffering mode.
- In cases where SetTemplatePrintArea has not been called, the entire print area will be cleared.

## Syntax

```
ErrorCode ClearTemplatePrintData
          (uint transactionNumber)
```

## Argument

transactionNumber:

Specifies the transaction number (ID) to set.



When executing at stations other than the physical endorsement station, please designate 0x40000000. If a value other than 0x40000000 is designated, the operation is definitely done to the physical endorsement station.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_NOT_EXEC	-470	Format-specified print mode has not been executed with TemplatePrint.
ERR_NOT_FOUND	-220	Data not found



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetTransactionNumber

Gets the currently set transaction number.



- The transaction number (ID) acquired by this API is the value used for the next scanning process, and it cannot be used as a parameter for GetMicrText and GetScanImage.
- When getting MICR text or a scan image with GetMicrText and GetScanImage, use the transaction number (ID) provided with the scanning status MF\_DATARECEIVE\_DONE or MF\_CHECKPAPER\_PROCESS\_DONE.

## Syntax

```
ErrorCode GetTransactionNumber  
(out int transactionNumber)
```

## Argument

transactionNumber:

Specifies the memory address where the transaction number (ID) is saved.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetTransactionNumber

Sets the transaction number.

---

## Syntax

```
ErrorCode SetTransactionNumber(int transactionNumber)
```

## Argument

transactionNumber:

Specifies the transaction number (ID) to set. The range of values that can be set is from 0 to 999999999.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetTransactionNumberWithIncremental

Sets the transaction number and the incremental value.

---

## Syntax

```
ErrorCode SetTransactionNumberWithIncremental  
        (int number, int increment)
```

## Argument

number:

Specifies the transaction number (ID) to set. The range of values that can be set is from 0 to 999999999.

increment:

Specifies the incremental value of transaction number to set. The range of values that can be set is from 1 to 999999999.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## SetWaterfallMode

Specifies the Waterfall mode.

### Syntax

```
ErrorCode SetWaterfallMode
    (WaterfallMode eWaterfallMode)
```

### Argument

eWaterfallMode:

Specifies the Waterfall mode.

Constant	Description
WATERFALL_MODE_DISABLE	Disables Waterfall mode.
WATERFALL_MODE_STANDARD	Prioritizes the main pocket as the pocket to eject document to. Ejects to the sub pocket when the main pocket is detected as being near full. When the main pocket near full status has been removed document will return to being ejected to the main pocket.
WATERFALL_MODE_INHERIT_POCKET	Prioritizes the pocket that document is ejected to when scanning completed previously as the pocket to eject document to. Switches pockets that document is ejected to when the pocket that document is ejected to is detected as being near full. The pocket that document is ejected to will not switch over when both pockets are detected as being near full.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetIQAResult

Gets IQA result.



- This API can be used when the SCNSetImageQuality settings are EPS\_BI\_SCN\_1BIT(Black and White) or EPS\_BI\_SCN\_8BIT(RGB grayscale).
- For card scans, you cannot get IQA results with this API. (ErrorCode:ERR\_NOT\_EXEC)

## Syntax

```
ErrorCode GetIqaResult(int dwTransactionNumber,  
                        MFIqaResult pResult)
```

## Argument

dwTransactionNumber:

Specifies the transaction number (ID) for the MICR text acquired.

pResult:

Specifies the memory address of the MFIqaResult class.

The following result is set for each IQA validation item.

Refer to [MFIqaResult Class - Properties \(p.242\)](#).

Constant	Description
IQARESULT_NOT_TESTED	IQA validation is not executed.
IQARESULT_PASS	Passed IQA validation.
IQARESULT_NOT_PASS	Not passed IQA validation.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_BUFFER_OVER_FLOW	-140	Buffer overflow error
ERR_NOT_FOUND	-220	No data error
ERR_NOT_EXEC	-470	Process not being executed
ERR_EXEC_FUNCTION	-310	Cannot be used due to another API being executed.
ERR_RESET	-400	Cannot be used because the device is being reset.



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetVersion

Acquires a driver version or module version.

## Syntax

```

ErrorCode GetVersion
    (DriverType eDriverType, VersionType eType
    , MFVersion ptVersion)
  
```

## Argument

**eDriverType:** The driver regarded as the acquisition object is designated.

Constant	Description
DRIVER_TYPE_S9000	TM-S9000 Driver
DRIVER_TYPE_S2000	TM-S2000 Driver

**eType:** The type of the version to be acquired. One of the following values can be specified.

Constant	Description
VERSION_TYPE_DRIVER	Driver version
VERSION_TYPE_USB	TMUSBDriver version
VERSION_TYPE_MICR	Magnetic waveform analysis module version
VERSION_TYPE_MICR_E13B	Magnetic waveform analysis module version(E13B)
VERSION_TYPE_MICR_CMC7	Magnetic waveform analysis module version(CMC7)
VERSION_TYPE_OCR	OCR recognition module version
VERSION_TYPE_IMAGE	Image processing module version
VERSION_TYPE_IQA	IQA module version
VERSION_TYPE_BARCODE	BARCODE module version
VERSION_TYPE_PH	Communication module version
VERSION_TYPE_MICR_PARSING	MICR parsing module version
VERSION_TYPE_BMPTORASTER	16-tone module version

**ptVersion:** Sets the address of the class for storing the version acquisition result.  
Refer to [MFVersion Class - Properties \(p.268\)](#).

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	No data error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNEnable

Set so that scanner extended functions can be used.

Before calling OpenMonPrinter, it is necessary to enable scanner extended functions by calling this argument.

---

## Syntax

```
static ErrorCode ESCNEnable(Storage storage)
```

## Argument

storage: Select a storing method for a cropped image stored using ESCNStoreImage.

Constant	Value	Description
CROP_STORE_MEMORY	0	Save in memory
CROP_STORE_FILE	1	Save in a file

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_PARAM	-90	Parameter error
ERR_ENABLE	-160	Cannot be used because OpenMonPrinter is called



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNGetAutoSize

Acquire the value of capAutoSize.

---

### Syntax

```
ErrorCode ESCNGetAutoSize(out AutoSize autoSize)
```

### Argument

autoSize: Select a memory address to set a capAutoSize value.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNSetAutoSize

Select the value of capAutoSize.

---

## Syntax

ErrorCode **ESCNSetAutoSize**(AutoSize autoSize)

## Argument

autoSize: Select a value for a capAutoSize.

Constant	Value	Description
CROP_AUTOSIZE_DISABLE	0	AutoSize processing disabled.
CROP_AUTOSIZE_ENABLE	1	AutoSize processing enabled.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNGetCutSize

Acquires a value of CutSize.

### Syntax

```
ErrorCode ESCNGetCutSize(out int cutSize)
```

### Argument

cutSize: Specify the memory address to store the CutSize value in increments of 0.1 mm.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNSetCutSize

Sets a value of CutSize.

---

## Syntax

```
ErrorCode ESCNSetCutSize(int cutSize)
```

## Argument

cutSize: Specify the width of left and right margins of an image data to be cropped out within a range of 0 to 1500 in increments of 0.1 mm. The default after executing OpenMonPrinter is zero.

cutSize	Description
0	No cropping operation is performed.
1 to 1500	The image is cropped to the specified size.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNGetRotate

Obtains the information whether the setting for rotating image data 90° has been made to the driver. Also, obtains the value of capRotate.

### Syntax

```
ErrorCode ESCNGetRotate(out Rotate rotate)
```

### Argument

rotate: Specify the address of the memory in which a return value of capRotate is stored.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNSetRotate

Specifies whether rotate the obtained image data 90° to the driver. Also sets a value to capRotate.



The set Rotate processing flag is not applied until the scanning of the next image data. (The Rotate process is not applied to the image data already scanned and stored in the WORD AREA.)

## Syntax

```
ErrorCode ESCNSetRotate(Rotate rotate)
```

## Argument

rotate: Specify the value for capRotate.

Constant	Value	Description
CROP_ROTATE_ENABLE	1	Processing Rotate enabled
CROP_ROTATE_DISABLE	0	Processing Rotate disabled

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNGetDeSkew

Obtains a threshold value of the skew angle currently set in the driver.

### Syntax

```
ErrorCode ESCNGetDeSkew(out ushort lpwAngle)
```

### Argument

lpwAngle: Specify the address of the memory to store the threshold value of the skew angle. (The unit of the argument is 0.01°)

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNSetDeSkew

Specifies a threshold value for the skew angle to execute DeSkew.

## Syntax

```
ErrorCode ESCNSetDeSkew(ushort wAngle)
```

## Argument

wAngle: Specify a threshold value for the skew angle. (unit: 0.01×)The following values can also be set.

Constant	Value	Description
DESKEW_ALL	0	Executes DeSkew.
DESKEW_DISABLE	65535	Does not execute DeSkew.

When a value other than DESKEW\_ALL or DESKEW\_DISABLE is specified and if the value is other than 1 - 8999, a parameter error occurs.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNGetDocumentSize

Acquire the values of documentWidth and documentHeight.



This API is compatible with the TM-J9000 API.

### Syntax

```
ErrorCode ESCNGetDocumentSize(out int documentWidth, out
                                int documentHeight)
```

### Argument

documentWidth:

Select a memory address to set a value of the width of the image data (unit: 0.1 mm).

documentHeight:

Select a memory address to set a value of the height of the image data (unit: 0.1 mm).

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNSetDocumentSize

Sets documentWidth and documentHeight of the image data.



- This API is compatible with the TM-J9000 API.
- The documentWidth and documentHeight settings made for this API are reflected on CROP\_AREA\_ENTIRE\_IMAGE of bCropArea, as specified with ESCNDefineCropArea.

## Syntax

```
ErrorCode ESCNSetDocumentSize(int documentWidth,  
                              int documentHeight)
```

## Argument

documentWidth:

This specifies the width of the image data (100 to 3000) in units of 0.1 mm.

documentHeight:

This specifies the height of the image data (100 to 3000) in units of 0.1 mm.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNDefineCropArea

Deletes the CropArea registration and the registered CropAreas.

## Syntax

ErrorCode **ESCNDefineCropArea**  
(byte areaNo, Rectangle area)

## Argument

areaNo: This specifies the CropAreaID (1 to 255) to be registered. This is a BYTE type. A user can use an ID from "2" to "255" for registering a CropArea.

Constant	Value	Description
CROP_AREA_RESET_ALL	0	All of the registered CropArea data is deleted.
CROP_AREA_ENTIRE_IMAGE	1	Zero is set for the CropArea start X and start Y coordinates, and the values of wdocumentWidth and wdocumentHeight are set for the end X and end Y coordinates.

startX: This specifies the start X coordinate of CropArea in units of 0.1 mm.  
 startY: This specifies the start Y coordinate (0 to documentHeight -1) of CropArea in units of 0.1 mm.  
 endX: This specifies the end X coordinate (value larger than startX)of CropArea in units of 0.1 mm.

Constant	Value	Description
CROP_AREA_RIGHT	65535	Sets the value of wdocumentWidth as the end X coordinate

endY: This specifies the end Y coordinate (value larger than startY)of CropArea in units of 0.1 mm.

Constant	Value	Description
CROP_AREA_BOTTOM	65535	This sets the value of wdocumentHeight as the end Y coordinate

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNGetMaxCropAreas

Acquires the maximum supported data count that can be registered for CropArea.

---

## Syntax

```
ErrorCode ESCNGetMaxCropAreas(out int count)
```

## Argument

count: This specifies the memory address in which the maximum supported data count that can be registered for CropArea is stored.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNStoreImage

Crops the CropArea specified with bCropAreaID from the image data in the work area, and then saves it to either a file or memory using the save method specified with ESCNEnable.

### Syntax

```
ErrorCode ESCNStoreImage(int fileIndex, String fileID,
    String imageTagData, byte cropAreaID)
```

### Argument

- fileIndex: This specifies FileIndex (an identifier) of the Crop image to be saved. The null can also be specified.
- fileID: This specifies FileID (an identifier) of the Crop image to be saved. This is an LPSTR type. A character string of up to 64 bytes can be specified. The null can also be specified. Note that none of \ / : ; \* ? " < > | can be used.
- imageTagData: This specifies ImageTagData (an identifier) of the Crop image to be saved. This is an LPSTR type. A character string of up to 64 bytes can be specified. The null can also be specified. Note that none of \ / : ; \* ? " < > | can be used.
- cropAreaID: This specifies the CropAreaID (1 to 255) registered with ESCNDefineCropArea.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Memory is insufficient
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_DISK_FULL	-170	There is insufficient free space on the disk
ERR_NO_IMAGE	-180	The image data does not exist
ERR_ENTRY_OVER	-190	It is not possible to register more than the maximum allowed number of items.
ERR_CROPAREAID	-200	The specified Crop Area does not exist
ERR_EXIST	-210	The specified data has already been saved
ERR_IMAGE_FILEOPEN	-230	Open failure
ERR_IMAGE_UNKNOW NFORMAT	-240	Format injustice
ERR_IMAGE_FAILED	-250	Image data creation failed
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNClearImage

Deletes the stored Crop image data.

## Syntax

```
ErrorCode ESCNClearImage(ClearImageFlag flag, int  
fileIndex, String fileID, String imageTagData)
```

## Argument

flag: This specifies the deletion method. By using the deletion methods appropriately, it is possible to specify the Crop image data to be deleted. Refer to the explanation.

Macro Definition (Constant)	Value	Description
CROP_CLEAR_ALL_IMAGE	0	Deletes all the Crop image save data
CROP_CLEAR_BY_FILEINDEX	1	Deletes the Crop image data specified by dwFileIndex
CROP_CLEAR_BY_FILEID	2	Deletes the Crop image data specified by pFileID
CROP_CLEAR_BY_IMAGETAGDATA	4	Deletes the Crop image data specified by pImageTagData

fileIndex: This specifies FileIndex (an identifier) of the Crop image data to be deleted.

fileID: This specifies FileID (an identifier) of the Crop image data to be deleted. Note that none of \ / : ; \* ? " < > | can be used.

imageTagData: This specifies ImageTagData (an identifier) of the Crop image data to be deleted. Note that none of \ / : ; \* ? " < > | can be used.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_NOT_FOUND	-220	No data error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## ESCNRRetrieveImage

Acquires the Crop image data that is saved to memory or a file.

### Syntax

- ErrorCode **ESCNRRetrieveImage**(int fileIndex, String fileID, String imageTagData, out byte[] data)
- ErrorCode **ESCNRRetrieveImage**(int fileIndex, String fileID, String imageTagData, out Image image)

### Argument

fileIndex:	This specifies FileIndex (an identifier) of the Crop image data to be acquired. While null can be specified, doing so prevents a search being made for the FileIndex.
fileID:	This specifies FileID (an identifier) of the Crop image data to be acquired. Note that none of \ / : ; * ? " < >   can be used. While null can be specified, doing so prevents a search being made for the FileID.
imageTagData:	This specifies ImageTagData (an identifier) of the Crop image data to be acquired. Note that none of \ / : ; * ? " < >   can be used. While null can be specified, doing so prevents a search being made for the ImageTagData.
data:	Image data as a byte array.
image:	Image data as an Image object.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_BUFFER_OVER_FLOW	-140	ERR_BUFFER_OVER_FLOW
ERR_NOT_FOUND	-220	No data error
ERR_IMAGE_FILEOPEN	-230	Open failure
ERR_IMAGE_FILEREAD	-290	Read of the image data file failed
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ESCNGetRemainingImages

Acquires the CropArea remaining count that can be registered.

---

## Syntax

```
ErrorCode ESCNGetRemainingImages(out int count)
```

## Argument

count: This specifies the memory address where the CropArea remaining count that can be registered is set.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetBehaviorToScnResult

Sets the behavior to the result for scanning.

## Syntax

```
ErrorCode SetBehaviorToScnResult
    (MfEject bEject, MfStamp bStamp,
    MfProcessContinue bNextCheck)
```

## Argument

bEject: Sets the ejection method of check sheets.

Constant	Description
MF_EJECT_MAIN_POCKET	Ejects into the main pocket.
MF_EJECT_SUB_POCKET	Ejects into the sub pocket.
MF_EJECT_NOEJECT	Does not eject.

bStamp: Sets whether to enable a franker.

Constant	Description
MF_STAMP_DISABLE	Does not perform franking.
MF_STAMP_ENABLE	Performs franking.



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

bNextCheck: Sets whether to feed the next check sheet when ejecting the current one.

Constant	Description
MF_PROCESS_CONTINUE_OVERLAP	Starts the next reading process while ejecting documents.
MF_PROCESS_CONTINUE_NOOVERLAP	Starts the next reading process after ejecting documents.
MF_PROCESS_CONTINUE_CANCEL	Cancel the next reading process.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Unsupported



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetNumberOfDocuments

The number of checks to be loaded is specified in the continuous scan execution API.

## Syntax

ErrorCode **SetNumberOfDocuments**(byte bNumber)

## Argument

bNumber: Specifies the number (0 to 100) of documents to read. The initial value is 0.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	A scan is being executed.
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## Description

- ❑ If 0 is specified for bNumber, all the documents set in the ASF (Auto Sheet Feeder) are read.
- ❑ If 1 or larger is specified for bNumber, reading ends when reading of the specified number of documents is complete. The setting made using this API is valid until CloseMonPrinter is run.
- ❑ If reading of the documents set in the ASF is completed before reading the number of documents specified using this API, the error code (ERR\_LESS\_CHECKS) is sent, indicating that the specified number of documents have not been read to SubStatus of the reading status MF\_FUNCTION\_DONE.



This API can be used to set the number of checks to be loaded only when MF\_ACTIVATE\_MODE\_HIGH\_SPEED is set for ActivationMode property in the MFProcess class that is specified with the continuous scan execution API. If MF\_ACTIVATE\_MODE\_CONFIRMATION is set, the setting specified with this API will be ignored. Application can control number of scanning document using [SetBehaviorToScnResult \(p.173\)](#).

## SelectErrorEjectAtContinuously

Sets the paper eject method used for when an error is detected while reading check paper continuously.



- This API is compatible with the TM-J9000 API.
- For TM-S9000/S2000, please use MFProcess class for setting paper eject method.

### Syntax

```
ErrorCode SelectErrorEjectAtContinuously
(ErrorEjectError error, ErrorEjectSetting
setting)
```

### Argument

**error:** Specifies the type of error for setting the paper eject method. The following values can be set individually or as a logical sum.

Constant	Description
EPS_BI_ERROREJECT_ERROR_ALL	All four of the following: double feed, magnetic waveform detection error, unrecognized character detection error, and noise error.
EPS_BI_ERROREJECT_ERROR_DOUBLEFEED	Double feed.
EPS_BI_ERROREJECT_ERROR_NODATA	Magnetic waveform detection error.
EPS_BI_ERROREJECT_ERROR_BADDATA	Unrecognized character detection error.
EPS_BI_ERROREJECT_ERROR_NOISE	Noise error.

**setting:** Specifies the paper eject method. One of the following values can be set.

Constant	Description
EPS_BI_ERROREJECT_SETTING_DISCHARGE	Paper ejected to main pocket.
EPS_BI_ERROREJECT_SETTING_RELEASE	Paper released at slip position.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SelectJamDetect

Sets the paper detection sensor used for paper jam detection.



This API is compatible with the TM-J9000 API.

## Syntax

ErrorCode **SelectJamDetect**(JamDetect jamDetect)

## Argument

jamDetect: Specifies the sensor for jam detect. One of the following values can be set.

Constant	Description
EPS_BI_JAM_DETECT_USE_UPPER_LEFT_SENSOR	An upper left paper sensor is used for paper jam detection. Although paper jam detection accuracy is high, when the check paper with which the upper left broke is scanned, it becomes a paper jam error.
EPS_BI_JAM_DETECT_NOT_USE_UPPER_LEFT_SENSOR	An upper left paper sensor is not used for paper jam detection. Although paper jam detection accuracy is a little inferior, the check paper with which the upper left broke can be scanned.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# MICRGetStatus

Acquires the MICR status.



- This API is compatible with the TM-J9000 API.
- Refer to [MICR Status \(p.279\)](#) regarding the acquired MICR status.

## Syntax

```
ErrorCode MICRGetStatus(out byte status)
```

## Argument

status: Specifies the memory address that acquires the MICR status.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_OFFLINE	-110	Cannot be used because of off-line return wait
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# MICRCleaning

Cleans the MICR mechanism.



If this function is called, the mechanism waits for the cleaning sheet to be inserted. Insert the cleaning sheet and carry out cleaning of the mechanism.

## Syntax

```
ErrorCode MICRCleaning()
```

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# InkHeadCleaning

Ink-jet head cleaning is executed.

The API finishes at the time when the cleaning command is sent, instead of waiting until the completion of head cleaning.

---

## Syntax

ErrorCode ***InkHeadCleaning()***

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# OpenDrawer

A drawer connected to the DKC (Drawer Kick Connector) is opened. Not supported in TM-S2000.

## Syntax

ErrorCode **OpenDrawer**(Drawer drawer, OpenPulse pulse)

## Argument

drawer: The drawer to be opened is selected.

Constant	Description
EPS_BI_DRAWER_1	Drawer 1 (3rd Drawer Kick Connector pin) will be opened.

pulse: Specifies the time when the drawer kick signal is on.

Constant	Description
EPS_BI_PLUSE_100	Signal for 100 milliseconds
EPS_BI_PLUSE_200	Signal for 200 milliseconds
EPS_BI_PLUSE_300	Signal for 300 milliseconds
EPS_BI_PLUSE_400	Signal for 400 milliseconds
EPS_BI_PLUSE_500	Signal for 500 milliseconds
EPS_BI_PLUSE_600	Signal for 600 milliseconds
EPS_BI_PLUSE_700	Signal for 700 milliseconds
EPS_BI_PLUSE_800	Signal for 800 milliseconds

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# RingBuzzer

Sounds a buzzer.

## Syntax

```
ErrorCode RingBuzzer(MfBuzzerTone bTone, byte bCount,
    ushort wOnTime, ushort wOffTime)
```

## Argument

bTone: Specifies the buzzer tone.

Constant	Description
MF_BUZZER_TONE_HIGH	High-pitched sound
MF_BUZZER_TONE_MIDDLE	Middle-pitched sound
MF_BUZZER_TONE_LOW	Low-pitched sound

bCount: Specifies the number of buzzers. The valid setting value is 1 to 8. When a value exceeding 8 is set, it will be rounded to 8.

wOnTime: Specifies the buzzer tone duration. The valid setting value is 100 to 800 (unit: mm). The specification must be made in 100 mm unit. A value less than 100 is rounded off to the nearest hundred. When a value exceeding 800 is set, it will be rounded to 800.

wOffTime: Specifies the off time of buzzer tone. The valid setting value is 100 to 800 (unit: mm). The setting must be made in 100 mm unit. A value less than 100 is rounded off to the nearest hundred. A value outside the valid range is rounded to the approximate value that can be specified. 0 (zero) can be set as well as valid values for the setting. When 0 is specified, the buzzer keeps sounding during the duration calculated by the following expression: Ring duration x Number of buzzers. No value is rounded to 0.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetCounter

Acquires the maintenance counter value.



Refer to [Maintenance Counter \(p.278\)](#) regarding the device counter and the acquired maintenance counters.

## Syntax

- `ErrorCode GetCounter(CounterIndex counter, bool cumulative, out int value)`

### Argument

counter: Enumeration of the possible counters.  
cumulative: If true, read cumulative value.  
value: Returns the maintenance counter.

- `ErrorCode GetCounter(byte counter, out int value)`

### Argument

counter: Value of the counter to read.  
value: Returns the maintenance counter.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ResetCounter

Acquires the maintenance counter value.



The maintenance counters include those that can be reset by the user and those that are not reset and count. Refer to [Maintenance Counter \(p.278\)](#).

## Syntax

- ErrorCode **ResetCounter**(CounterIndex counter)

Argument

counter: Enumeration of the possible counters.

- ErrorCode **ResetCounter**(byte counter)

Argument

counter: Value of the counter to read.

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_OFFLINE	-110	Cannot be used because of off-line return wait
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# LoadAPISettings

Loads the API Setting File to set the driver.

(Refer to the TM-S9000/S2000 API Reference Guide for Win32/64 - Chapter6 "Setting File".)

## Syntax

```
ErrorCode LoadAPISettings(String SettingsFilePath)
```

## Argument

SettingsFilePath:

Specifies the path to the API Setting File.

Value	Description
Any string	The full path to the file must be specified.
APISETTINGS_DEFAULT	If APISETTINGS_DEFAULT is specified, it is assumed that APISettings.ini located in the same folder as the driver is specified.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	File not found
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetConfigure

Set enable/disable for 2nd receipt, machine tape, etc.

## Syntax

```
ErrorCode SetConfigure
    (ConfigID configID, ConfigType configType)
```

## Argument

configID: Specify the function to set.

Value	Description
MF_CONFIG_ID_2ND_RECEIPT (Not supported in TM-S2000)	Settings for the 2nd receipt function
MF_CONFIG_ID_SCAN_LENGTH	Settings for scan paper length
MF_CONFIG_ID_MICR_RECOG	Settings for MICR recognition type
MF_CONFIG_ID_ENDORSE_PRINT_POSITION	Settings for endorsement print position

configType: Settings for functions specified in ConfigID

Function	Value	Description
2nd Reciept (Not supported in TM-S2000)	MF_CONFIG_TYPE_2ND_RECEIPT_DISABLE	DISABLE
	MF_CONFIG_TYPE_2ND_RECEIPT_ENABLE	ENABLE
Machine tape	MF_CONFIG_TYPE_SCAN_LENGTH_NORMAL	Normal paper length
	MF_CONFIG_TYPE_SCAN_LENGTH_EXPAND	Expanded paper length (for machine tape)
MICR recognition type	MF_CONFIG_TYPE_MICR_RECOG_ANSI	ANSI
	MF_CONFIG_TYPE_MICR_RECOG_CCCC	C&CCC
Endorsement print position	MF_CONFIG_TYPE_ENDORSE_PRINT_POSITION_NORMAL	Print starting position is based on the leading edge of the paper
	MF_CONFIG_TYPE_ENDORSE_PRINT_POSITION_PAYEE_ENDORSE	Print starting position is based on the back-end of the paper

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# ResetPrinter

Resets the device whose status is being monitored.

If an error occurs in the device, resolve the cause of the error, and then reset the device.



- Use this API when a recoverable error occurs. Refer to [Device Status \(p.274\)](#) for details on recoverable errors.
- Cancels print jobs when this is called while printing.

---

## Syntax

```
ErrorCode ResetPrinter()
```

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	A time out error occurred
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# CancelError

Restores recoverable device errors.

---

## Syntax

ErrorCode **CancelError**()

Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_TIMEOUT	-70	A time out error occurred
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# GetType

Acquires the device type ID.



Refer to [Type ID \(p.294\)](#).

## Syntax

```
ErrorCode GetType(out byte typeid, out byte font, out  
byte exrom, out byte euspecial)
```

## Argument

typeid: Returns the device type ID.  
font: Not used  
exrom: Not used  
euspecial: Not used

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_ACCESS	-80	Reading/writing with the device is not possible (printing in progress)
ERR_PARAM	-90	Parameter error
ERR_OFFLINE	-110	Cannot be used because of off-line return wait
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

## GetOfflineCodeByIndex

Acquires the cause of the device to go offline from the leading byte.



- The cause for going offline is in the device set in the API settings file. For more information, refer to Chapter 6 "Setting File" in the TM-S9000/S2000 API Reference Guide for Win32/64.
- Data obtainable by this API is a device-specific value and may differ from the data for the previous models.
- Refer to [Offline Code \(p.280\)](#).

### Syntax

```
ErrorCode GetOfflineCodeByIndex
(int iIndex, out byte lpbOfflinecode)
```

### Argument

iIndex: Specifies bytes (1 to 10) of the acquired cause of the device going offline.  
lpbOfflinecode: Returns the 5-byte value indicating the reason for the device going offline.



When the cause of the device going offline is acquired while the device is online, zero will be written to the leading byte of lpOfflinecode.

### Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_RESET	-400	Cannot be used because the device is being reset



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SCNGetClampStatus

Acquires the clamp status of cut sheets.



This API is compatible with the TM-J9000 API. Since there is no clamp status information in TM-S9000 or TM-S2000, "01000000(0x40)" is always returned.

## Syntax

ErrorCode **SCNGetClampStatus**(out byte status)

## Argument

status: The clamp status(01000000 (0x40)) is set.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	The handle value that specifies the device is incorrect
ERR_PARAM	-90	Parameter error



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# SetPaperThickness

This function does not work with TM-S9000 and TM-S2000, because this function is just for compatibility with TM-S1000 API.

## Syntax

```
ErrorCode SetPaperThickness(ushort wThreshold)
```

## Argument

wThreshold: Specifies the double feed threshold. The valid specification range is 1 to 40 (0.01 mm to 0.40 mm). When "0" is specified, the threshold value when calling OpenMonPrinter is applied.

## Return value

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed



For a list of return values and troubleshooting actions, refer to [Return value \(p.43\)](#).

# Properties

This sections contains descriptions of all of the properties exposed by the MFDevice class.

## LastError

Retrieves the return value after executing an API.

### Type

Property: Read only  
Data: Enum ErrorCode

### Values

Refer to [Return value \(p.43\)](#) for details.

## IsValid

Retrieves the result of executing OpenMonPrinter.

### Type

Property: Read only  
Data: bool

### Values

Value	Description
true	OpenMonPrinter succeeded
false	OpenMonPrinter failed

## Status

Retrieves the current device status.

### Type

Property: Read only  
Data: Enum ASB

### Values

Refer to [Device Status \(p.274\)](#) for details.

## InkStatus

Retrieves the current ink status.

### Type

Property: Read only  
Data: Enum InkASB

### Values

Refer to [Ink Status \(p.277\)](#) for details.

## OfflineCode

Retrieves the offline cause.

### Type

Property: Read only  
Data: byte(5)

### Values

Refer to [Offline Code \(p.280\)](#) for details.

## ESCNAutoSize

Set/Get image auto size setting value.

### Type

Property: Read/Write  
Data: Enum AutoSize

### Values

Value	Description
CROP_AUTOSIZE_DISABLE	AutoSize processing disabled.
CROP_AUTOSIZE_ENABLE	AutoSize processing enabled.

## ESCNCutSize

Set/Get image cut size.

### Type

Property: Read/Write  
Data: int

### Values

1 to 1500 (unit: 0.1 mm)

## ESCNRotate

Set/Get image rotation setting value.

### Type

Property: Read/Write  
Data: Enum Rotate

### Values

Value	Description
CROP_ROTATE_ENABLE	Processing Rotate enabled
CROP_ROTATE_DISABLE	Processing Rotate disabled

## ESCNDocumentSize

Set/Get size of document to scan.

### Type

Property: Read/Write  
Data: System.Drawing.Size

### Values

Width/Height of the image data. (unit: 0.1 mm)

## ESCNDDeSkew

Set/Get image skew angle setting value.

### Type

Property: Read/Write  
Data: ushort

### Values

Skew angle threshold value. (unit: 0.01 degree)

## ESCNRemainingImages

Get the remaining number of data of the Crop image that can be registered.

### Type

Property: Read only  
Data: int

### Values

0 to 255

## TransactionNumber

Set/Get the transaction number.

### Type

Property: Read/Write  
Data: int

### Values

0 to 999999999.

## TransactionIncrement

Set count up the number of transaction number.

### Type

Property: Write only  
Data: int

### Values

0 to 999999999.

## PrintStation

Set/Get the station for printing.

### Type

Property: Read/Write  
Data: Enum PrintStation

### Values

Value	Description
MF_ST_ROLLPAPER (Not supported in TM-S2000)	Sets roll paper as the station for printing.
MF_ST_VALIDATION	Sets validation as the station for printing.
MF_ST_E_ENDORSEMENT	Sets electronic endorsement as the station for printing.
MF_ST_E_ENDORSEMENT_BACK	Sets back face electronic endorsement as the station for printing. (Same as MF_ST_E_ENDORSEMENT.)
MF_ST_E_ENDORSEMENT_FRONT	Sets front face electronic endorsement as the station for printing.
MF_ST_PHYSICAL_ENDORSEMENT	Sets physical endorsement as the station for printing.
MF_ST_FEEDER	Sets U-shaped mechanism as the station for printing.

## PrintSize

Set/Get the printing image size.

### Type

Property: Read/Write  
Data: System.Drawing.Size

### Values

Width/Height of the printing image data size. (unit: mm)

## PrintPosition

Set/Get the printing start position.

### Type

Property: Read/Write  
Data: System.Drawing.Point

### Values

Horizontal/Vertical location of print. (unit: mm)

## PrintAlignment

Set/Get the station for printing.

### Type

Property: Read/Write  
Data: enum PrintAlignment

### Values

Value	Description
MF_PRINT_ALIGNMENT_LEFT	Prints with left alignment.
MF_PRINT_ALIGNMENT_CENTER	Prints with center alignment.
MF_PRINT_ALIGNMENT_RIGHT	Prints with right alignment.

# Events

This section contains descriptions of all of the events exposed by the MFDevice class.

## StatusCallback/StatusCallbackEx

Event when the status of the device status is changed.

### Syntax

- **event StatusCallbackHandler StatusCallback**

Delegate

***delegate void StatusCallbackHandler(ASB asb)***

Argument

asb: The device status is set. Refer to [Device Status \(p.274\)](#).

- **event StatusCallbackHandlerEx StatusCallbackEx**

Delegate

***delegate void StatusCallbackHandlerEx  
(ASB asb, String portName)***

Argument

asb: The device status is set. Refer to [Device Status \(p.274\)](#).

portName: Port to which the device is connected.

## InkStatusCallback/InkStatusCallbackEx

Event when the status of the ink status is changed.

---

### Syntax

- **event InkStatusCallbackHandler InkStatusCallback**

Delegate

**delegate void InkStatusCallbackHandler(InkASB ink)**

Argument

ink: The ink status is set. Refer to [Ink Status \(p.277\)](#).

- **event InkStatusCallbackHandlerEx InkStatusCallbackEx**

Delegate

**delegate void InkStatusCallbackHandlerEx  
(InkASB ink, String portName)**

Argument

ink: The ink status is set. Refer to [Ink Status \(p.277\)](#).

portName: Port to which the device is connected.

## MfDone

Event when SCNMICRFunction operation is complete.

---

### Syntax

**event MfDoneHandler MfDone**

Delegate

**delegate void MfDoneHandler()**

## SCNMICRStatusCallback

Event when the stage of SCNMICRFunctionContinuously or SCNMICRFunctionPostPrint operation changes.

### Syntax

**event SCNMICRStatusCallbackHandler SCNMICRStatusCallback**

Delegate

**delegate void SCNMICRStatusCallbackHandler  
(int TransactionNumber, MainStatus mainStatus  
, ErrorCode subStatus, String portName)**

Argument

TransactionNumber:	The transaction number (ID) corresponding to the check sheet of the processing status source.
mainStatus:	The main status of the processing status. For details of the processing status, refer to <a href="#">Processing status list (p.78)</a> .
subStatus:	The sub status of the processing status. For details of the processing status, refer to <a href="#">Processing status list (p.78)</a> .
portName:	Port to which the device is connected.

## StartEndorsementStatusCallback

This event function is invoked when the device is ready to receive endorsement print data.

### Syntax

**event StartEndorsementStatusCallbackHandler  
StartEndorsementStatusCallback**

Delegate

**delegate void StartEndorsementStatusCallbackHandler  
(int TransactionNumber, String portName)**

Argument

TransactionNumber:	Transaction number (ID) corresponding to the check sheet from which the processing status has been issued.
portName:	Port to which the device is connected.

## EndEndorsementStatusCallback

This event is invoked when the device is no longer capable of receiving endorsement print data.

---

### Syntax

```
event EndEndorsementStatusCallbackHandler  
EndEndorsementStatusCallback
```

Delegate

```
delegate void EndEndorsementStatusCallbackHandler  
(int TransactionNumber, String portName)
```

Argument

TransactionNumber:

Transaction number (ID) corresponding to the check sheet from which the processing status has been issued.

portName: Port to which the device is connected.

## MFBase Class - Properties

This sections contains descriptions of all of the properties exposed by the MFBase class.

### Version

This is the class version.

#### Type

Property:	Read only
Data:	int

#### Values

The version of class used internally within this class.

### Ret

The return value for this function is set.

#### Type

Property:	Read only
Data:	ErrorCode

#### Values

Refer to [MFBase.Ret \(p.76\)](#).

### Timeout

#### Type

Property:	Read/Write
Data:	int

#### Values

The time to wait before paper insertion is specified in seconds. The values that can be specified are 0 to 300 (five minutes), and when 0 is specified, there is no timeout. When there is a timeout, ERR\_PAPERINSERT\_TIMEOUT is returned. Timeouts other than paper insertion cannot be specified.

## ErrorEject

This property is used for compatibility. Set the paper ejection method with [MFProcess Class - Properties \(p.220\)](#).

### Type

Property: Read/Write  
Data: MfEjectType

### Values

MfEjectType	Description
MF_EJECT_DISCHARGE	Specifies that the paper should be ejected to a pocket.
MF_EJECT_RELEASE	Specifies that the paper should be unclamped and left in the same location.

## SuccessEject

This property is used for compatibility. Set the paper ejection method with [MFProcess Class - Properties \(p.220\)](#).

### Type

Property: Read/Write  
Data: MfEjectType

### Values

MfEjectType	Description
MF_EJECT_DISCHARGE	Specifies that the paper should be ejected to a pocket.
MF_EJECT_RELEASE	Specifies that the paper should be unclamped and left in the same location.

## UseNVMemory

Not used.

## BuzzerHz[3]

This sets the frequency of the buzzer for MICR reading.

### Type

Property: Read/Write  
Data: *BuzzerHzClass.MFBuzzerCount*

### Indexer

MfBuzzerType	Description
MF_BUZZER_TYPE_SUCCESS	Successful read of check.
MF_BUZZER_TYPE_ERROR	Error occurred while reading check.
MF_BUZZER_TYPE_WFEED	Multiple check feed detected.

### Values

MfBuzzerHz	Description
MF_BUZZER_HZ_440	Set buzzer frequency to 440Hz
MF_BUZZER_HZ_880	Set buzzer frequency to 880Hz
MF_BUZZER_HZ_4000	Set buzzer frequency to 4000Hz

## BuzzerCount[3]

This sets the number of times the buzzer corresponding to each element in bBuzzerHz sounds.

### Type

Property: Read/Write  
Data: *BuzzerHzClass.MFBuzzerCount*

### Indexer

MfBuzzerType	Description
MF_BUZZER_TYPE_SUCCESS	Successful read of check.
MF_BUZZER_TYPE_ERROR	Error occurred while reading check.
MF_BUZZER_TYPE_WFEED	Multiple check feed detected.

### Values

MfBuzzerHz	Description
MF_BUZZER_HZ_440	Set buzzer frequency to 440Hz
MF_BUZZER_HZ_880	Set buzzer frequency to 880Hz
MF_BUZZER_HZ_4000	Set buzzer frequency to 4000Hz

---

## PortName

This sets the port name.

### Type

Property: Read only

Data: String

### Values

The port name associated with this device.

## MFMicr Class - Properties

This sections contains descriptions of all of the properties exposed by the MFMicr class.

### Version

This is the class version.

#### Type

Property: Read only  
Data: int

#### Values

The version of class used internally within this class.

### Ret

The return value for this function is set.

#### Type

Property: Read only  
Data: ErrorCode

#### Values

Refer to [MFMicr.Ret \(p.77\)](#).

### Font

Specifies the font of the MICR data to be read.

#### Type

Property: Read/Write  
Data: MfMicrFont

#### Values

MfMicrFont	Description
MF_MICR_FONT_E13B	Read MICR characters with the E13B font.
MF_MICR_FONT_CMC7	Read MICR characters with the CMC7 font.

### MicrOcrSelect

This member is compatible with the TM-S1000 API. With the TM-S9000/S2000 API, the settings for this member are ignored and MF\_MICR\_USE\_BOTH is always applied.

## Parsing

When TRUE is specified, parsing is carried out, and the character string is set in szAccountNumber, szAmount, szBankNumber, szSerialNumber, szEPC and szTransitNumber.

### Type

Property: Read/Write  
Data: bool

### Values

bool	Description
true	Parse magnetically read characters.
false	Do not parse magnetically read characters.

## Status

The MICR reading status is set.

### Type

Property: Read only  
Data: byte

### Values

Bit	Function	Value	
		0	1
0	Reading font	E13B	CMC7
1, 2	Reserved	Fixed to 0	
3	Detailed information	-	Added
4	Reread	-	Disabled
5	Reading results	Success	Failure
6	OCR processing error	No	Yes
7	Readout data reception error	No	Yes

## Detail

Detailed MICR reading status is set.

### Type

Property: Read only

Data: byte

### Values

Value	Information
40h	Success
41h	Check sheet reading has not ever been executed. (The SCNMICRFunction function has not been invoked.)
44h	A delivery error occurred in the processing before reading.
45h	A magnetic waveform cannot be detected.
46h	Characters that cannot be analyzed were detected in the analysis processing.
47h	A double-feeding error or an insertion direction error occurred during check sheet reading.
48h	An abnormality was detected in noise measurement.
49h	Check sheet reading was stopped due to a feeding error.
4Bh	In reading, an error of paper length too long occurred.

## MicrStr

The read MICR data is set.

Refer to [MICR Control Characters \(p.208\)](#) for details on MICR control characters.

### Type

Property: Read only

Data: String

## MICR Control Characters

MICR data that can be retrieved includes control characters in addition to numbers. See below:

### E13B Font

MICR Character	Name	Alternate Character
	Transit	†
	Amount	α
	On-Us	ο
	Dash	-

### CMC7 Font

MICR Character	Alternate Character
	/
	#
	=
	>
	^

## AccountNumber

AccountNumber property value.

### Type

Property: Read only

Data: String

## Amount

Amount property value.

### Type

Property: Read only

Data: String

## BankNumber

BankNumber property value.

### Type

Property:	Read only
Data:	String

## SerialNumber

SerialNumber property value.

### Type

Property:	Read only
Data:	String

## EPC

EPC property value.

### Type

Property:	Read only
Data:	String

## TransitNumber

TransitNumber property value.

### Type

Property:	Read only
Data:	String

## CheckType

CheckType property value.

### Type

Property:	Read only
Data:	Check

## OcrReliableInfo.FirstSelectString

Best guess raw data read optically from MICR characters.

### Type

Property: Read only  
Data: String

## OcrReliableInfo.SecondSelectString

Second guess raw data read optically from MICR characters.

### Type

Property: Read only  
Data: String

## OcrReliableInfo.FirstSelectPercentage

Confidence level of best guess raw data read optically from MICR characters.

### Type

Property: Read only  
Data: int()

## OcrReliableInfo.SecondSelectPercentage

Confidence level of second guess raw data read optically from MICR characters.

### Type

Property: Read only  
Data: int()

## OnUSField

The MICR On-US field is stored.

### Type

Property: Read only  
Data: String

## AuxiliaryOnUSField

The Auxiliary On-US field is stored.

### Type

Property:	Read only
Data:	String

# MFScan Class - Properties

This sections contains descriptions of all of the properties exposed by the MFScan class.

## Version

This is the class version.

### Type

Property:	Read only
Data:	int

### Values

The version of class used internally within this class.

## Ret

The return value for this function is set.

### Type

Property:	Read only
Data:	ErrorCode

### Values

Refer to [MFScan.Ret \(p.76\)](#).

## ImageID

This specifies the ID for an image that is read.

### Type

Property:	Read/Write
Data:	int

## Resolution

This specifies the resolution for an image that is read. Select one from the following preset parameters.

### Type

Property: Read/Write  
Data: MfScanDpi

### Values

MfScanDpi	Description
MF_SCAN_DPI_DEFAULT	Scan with default DPI (200 dpi)
MF_SCAN_DPI_100	Scan image at 100 DPI
MF_SCAN_DPI_120	Scan image at 120 DPI
MF_SCAN_DPI_200	Scan image at 200 DPI
MF_SCAN_DPI_240	Scan image at 240 DPI
MF_SCAN_DPI_300	Scan image at 300 DPI
MF_SCAN_DPI_600	Scan image at 600 DPI

## AddInfoData

This specifies the size of additional character data.

### Type

Property: Read/Write  
Data: byte[]

## Image

Scanned image interpreted as a System.Drawing.Bitmap object.

### Type

Property: Read only  
Data: System.Drawing.Bitmap

## Data

Scanned image interpreted as a System.IO.Stream object.

### Type

Property: Read/Write  
Data: System.IO.Stream

## Status

This sets the status when the scan is completed.

### Type

Property: Read only  
Data: byte

### Values

Bit	Status Contents	ON / OFF	Value	Status
0	-	-	0x0000	Reserved (Fixed to 0)
1	-	-	0x0000	Reserved (Fixed to 0)
2	-	-	0x0000	Reserved (Fixed to 0)
3	Scanning face	ON	0x0008	Back
		OFF	0x0000	Front
4	Rescanned	ON	0x0010	Disabled
		OFF	0x0000	-
5	Scanning results	ON	0x0020	Abnormal end
		OFF	0x0000	Success
6	Scanning data overflow	ON	0x0040	No overflow
		OFF	0x0000	Not (Fixed)
7	Scanning data translation error	ON	0x0080	Ends with an error
		OFF	0x0000	No error

## Detail

This sets the detailed status when the scan is completed.

### Type

Property: Read only  
Data: byte

### Values

Value	Information
40h	Success
41h	No image reading result
42h	Cancellation of paper insertion waiting (Executing SCNMICRCancelFunction)
44h	Cancellation of image reading due to feed error
45h	Occurrence of double feed or insertion orientation error during image reading
46h	Detection of sending error before reading starts
48h	Detection of paper length error during reading

## MFDeviceFont Class - Constructor

Constructor and instance of the MFDeviceFont class, setting font parameters if supplied.

---

### Syntax

- **MFDeviceFont** (FontType font, FontScale size, UnderlineType underline, bool bold, PrintColor color, bool reverse)
- **MFDeviceFont** (FontType font, FontScale size, UnderlineType underline, bool bold, PrintColor color)
- **MFDeviceFont** (FontType font, FontScale size, UnderlineType underline, bool bold)
- **MFDeviceFont** (FontType font, FontScale size, UnderlineType underline)
- **MFDeviceFont** (FontType font, FontScale size, bool bold, PrintColor color, bool reverse)
- **MFDeviceFont** (FontType font, FontScale size, bool bold, PrintColor color)
- **MFDeviceFont** (FontType font, FontScale size, bool bold)
- **MFDeviceFont** (FontType font, FontScale size, PrintColor color, bool reverse)
- **MFDeviceFont** (FontType font, FontScale size, PrintColor color)
- **MFDeviceFont** (FontType font, FontScale size)
- **MFTrueType** ()

### Argument

font:	Refer to <a href="#">Font (p.216)</a> .
size:	Refer to <a href="#">Size (p.216)</a> .
underline:	Refer to <a href="#">Underline (p.216)</a> .
bold:	Refer to <a href="#">Bold (p.217)</a> .
color:	Refer to <a href="#">Color (p.217)</a> .
reverse:	Refer to <a href="#">Reverse (p.219)</a> .

# MFDeviceFont Class - Properties

This sections contains descriptions of all of the properties exposed by the MFDeviceFont class.

## Font

Set/Get the type of font.

### Type

Property: Read/Write  
Data: Enum FontType

### Values

Enum FontType	Value	Description
MF_PRINT_FONT_A	1	Device internal font A
MF_PRINT_FONT_B	2	Device internal font B

## Size

Set/Get the font size.

### Type

Property: Read/Write  
Data: Enum FontScale

### Values

Enum FontScale	Value	Description
MF_PRINT_FONT_W1_H1	0x0011	Normal size
MF_PRINT_FONT_W1_H2	0x0012	Extra tall font (double height)
MF_PRINT_FONT_W2_H1	0x0021	Extra wide font (double width)
MF_PRINT_FONT_W2_H2	0x0022	Double size font (double height and double width)

## Underline

Set/Get the underline setting.

### Type

Property: Read/Write  
Data: Enum UnderlineType

### Values

Enum UnderlineType	Value	Description
MF_PRINT_UNDERLINE_1	0x00000010	Single width underline
MF_PRINT_UNDERLINE_2	0x00000020	Double width underline
MF_PRINT_NO_UNDERLINE	-	No underline

## Bold

Set/Get the bold setting.

### Type

Property: Read/Write  
Data: bool

### Values

Value	Description
true	Bold typeface weight
false	Normal typeface weight

## Color

Set/Get the color setting.

### Type

Property: Read/Write  
Data: Enum PrintColor

### Values

Enum PrintColor	Value	Description
MF_PRINT_BLACK	0x00000100	Primary color
MF_PRINT_1ST_COLOR	0x00000100	Primary color
MF_PRINT_COLOR	0x00000200	Secondary color

## Reverse

Set/Get the reverse printing setting.

### Type

Property: Read/Write  
Data: bool

### Values

Value	Description
true	Typeface is reversed
false	Typeface is not reversed

## MFTTrueType Class - Constructor

Constructor and instance of the MFTTrueType class, setting font parameters if supplied.

---

### Syntax

- **MFTTrueType**(System.Drawing.Font font, PrintColor color, bool reverse)
- **MFTTrueType**(System.Drawing.Font font, PrintColor color)
- **MFTTrueType**(System.Drawing.Font font)
- **MFTTrueType**()

### Argument

font:	Refer to <a href="#">Font (p.219)</a> .
color:	Refer to <a href="#">Color (p.219)</a> .
reverse:	Refer to <a href="#">Reverse (p.219)</a> .

## MFTrueType Class - Properties

This sections contains descriptions of all of the properties exposed by the MFTrueType class.

### Font

A System.Drawing.Font object, the font object's Name, SizeInPoints and Bold and Underline properties are used and maintained.



All properties of the SystemDrawing.Font object are ignored other than Name, SizeInPoints, Bold and Underline.

#### Type

Property: Read/Write  
Data: System.Drawing.Font

### Color

Set/Get the color setting.

#### Type

Property: Read/Write  
Data: Enum PrintColor

#### Values

Enum PrintColor	Value	Description
MF_PRINT_BLACK	0x00000100	Primary color
MF_PRINT_1ST_COLOR	0x00000100	Primary color
MF_PRINT_COLOR	0x00000200	Secondary color

### Reverse

Set/Get the reverse printing setting.

#### Type

Property: Read/Write  
Data: bool

#### Values

Value	Description
true	Typeface is reversed
false	Typeface is not reversed

# MFProcess Class - Properties

This sections contains descriptions of all of the properties exposed by the MFProcess class.

## Version

This is the class version.

### Type

Property: Read only  
Data: int

### Values

The version of class used internally within this class.

## ActivationMode

Set/Get the activation mode for scanning.

### Type

Property: Read/Write  
Data: enum MfActivateMode

### Values

enum MfActivateMode	Description
MF_ACTIVATE_MODE_HIGH_SPEED	Execute processing in High Speed mode
MF_ACTIVATE_MODE_CONFIRMATION	Execute processing in Confirmation mode



- When MF\_ACTIVATE\_MODE\_HIGH\_SPEED is specified, the driver carries out all the error verification at scanning. The driver deals with errors according to the pre-set actions.
- The actions at error occurrence are specified in this class. Scan speed may slow down depending on the setting for the action at error occurrence.
- When MF\_ACTIVATE\_MODE\_CONFIRMATION is specified, the application carries out all the error verification at scanning. For details of the error verification by the application, refer to the function [SetBehaviorToScnResult \(p.173\)](#).

## PaperType

Set/Get the paper type for a scan target.

### Type

Property: Read/Write  
Data: enum MfPaperType

### Values

enum MfPaperType	Description
MF_PAPER_TYPE_OTHER	Scan check paper, stub or any other type of paper
MF_PAPER_TYPE_CHECK	Papers other than check sheets included



This property is compatible with the TM-S1000 API. With the TM-S9000/S2000 API, the setting for this property is ignored.

## StartWaitTime

Specify a time interval from when paper on the ASF is detected and scanning the paper is started.

### Type

Property: Read/Write  
Data: uint

### Values

Settable values are 0 to 6400 in units of ms. The default is zero.  
A value larger than 6400 is rounded down to 6400.

## SuccessStamp

Set/Get the whether to enable a Franker when scan completes successfully.

### Type

Property: Read/Write  
Data: enum MfStamp

### Values

enum MfStamp	Description
MF_STAMP_ENABLE	Enable a stamp when scan is completed successfully
MF_STAMP_DISABLE	Disable a stamp when scan is completed successfully



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## PaperMisInsertionErrorSelect

Set/Get the whether to detect the insertion direction error.

### Type

Property: Read/Write  
Data: enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	No error detected
MF_ERROR_SELECT_DETECT	Error detected

## PaperMisInsertionErrorEject

Set/Get the ejection method for when the insertion direction error is detected.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## PaperMisInsertionStamp

Set/Get the whether to enable a franker for when the insertion direction error is detected.

### Type

Property: Read/Write  
Data: enum MfStamp

### Values

enum MfStamp	Description
MF_STAMP_ENABLE	Franker enabled
MF_STAMP_DISABLE	Franker disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## PaperMisInsertionCancel

Set/Get the whether to continue reading operation when an insertion direction incorrect error occurs.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

## NoiseErrorSelect

Set/Get the whether to detect the external noise error.

### Type

Property: Read/Write  
Data: enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	No error detected
MF_ERROR_SELECT_DETECT	Error detected

## NoiseErrorEject

Set/Get the ejection method for when the external noise error is detected.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## NoiseStamp

Set/Get the whether to enable a franker for when the external noise error is detected.

### Type

Property: Read/Write  
Data: enum MfStamp

### Values

enum MfStamp	Description
MF_STAMP_ENABLE	Franger enabled
MF_STAMP_DISABLE	Franger disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## NoiseCancel

Set/Get the whether to continue reading operation when an external noise error occurs.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

## DoubleFeedErrorSelect

Set/Get the whether to detect the double feed error.

### Type

Property: Read/Write  
Data: enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	No error detected
MF_ERROR_SELECT_DETECT	Error detected

## DoubleFeedErrorEject

Set/Get the ejection method for when the double feed error is detected.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## DoubleFeedStamp

Set/Get the whether to enable a franker for when the double feed error is detected.

### Type

Property: Read/Write  
Data: enum MfStamp

### Values

enum MfStamp	Description
MF_STAMP_ENABLE	Franger enabled
MF_STAMP_DISABLE	Franger disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## DoubleFeedCancel

Set/Get the whether to continue reading operation when an double feed error occurs.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

## BaddataErrorSelect

Sets whether to detect unrecognizable MICR characters as errors.

When value of this property is MF\_ERROR\_SELECT\_DETECT and number of unrecognizable characters is greater than the permissible value set with BaddataCount, unrecognizable MICR characters are treated as an error.

### Type

Property: Read/Write  
Data: enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	No error detected
MF_ERROR_SELECT_DETECT	Error detected

## BaddataCount

Set the amount of unrecognizable characters permitted (the number below which is not determined as an error) when detecting unrecognizable MICR characters as errors.

When value of BaddataErrorSelect is MF\_ERROR\_SELECT\_DETECT and number of unrecognizable characters is greater than the permissible value set with this property, unrecognizable MICR characters are treated as an error.

### Type

Property: Read/Write  
Data: byte

### Values

Specifying zero allows no unanalyzable characters.

When a number (1 to 254) is specified, the specified number of unanalyzable characters are allowed.

## BaddataErrorEject

Set/Get the ejection method for when the MICR character recognition error is detected.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## BaddataStamp

Set/Get the whether to enable a franker for when the MICR character recognition error is detected.

### Type

Property: Read/Write  
Data: enum MfStamp

### Values

enum MfStamp	Description
MF_STAMP_ENABLE	Franker enabled
MF_STAMP_DISABLE	Franker disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## BaddataCancel

Set/Get the whether to continue reading operation when an MICR character recognition error occurs.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

## NodataErrorSelect

Set/Get the whether to detect errors when magnetic waveform is not found.

### Type

Property: Read/Write  
Data: enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	No error detected
MF_ERROR_SELECT_DETECT	Error detected

## NodataErrorEject

Set/Get the ejection method for when an error is detected because MICR magnetic waveform is not found.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## NodataStamp

Set/Get the whether to enable a franker for when an error is detected because MICR magnetic waveform is not found.

### Type

Property: Read/Write  
Data: enum MfStamp

### Values

enum MfStamp	Description
MF_STAMP_ENABLE	Franker enabled
MF_STAMP_DISABLE	Franker disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## NodataCancel

Set/Get the whether to cancel the action for when an error is detected because that MICR magnetic waveform is not found.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

## NearFullSelect

Set/Get the whether to permit scanning when the eject pocket is nearly full.

### Type

Property: Read/Write  
Data: enum MfNearFullSelect

### Values

enum MfNearFullSelect	Description
MF_NEARFULL_NOT_PERMIT	Scan not permitted
MF_NEARFULL_MAIN_PERMIT	Scan permitted(Main pocket)
MF_NEARFULL_SUB_PERMIT	Scan permitted(Sub pocket)
MF_NEARFULL_PERMIT	Scan permitted

## ResultPartialData

Set/Get the whether to acquire data that is being scanned when an error that does not interrupt any operation occurs in the middle of the scanning.

### Type

Property: Read/Write  
Data: enum MfResult

### Values

enum MfResult	Description
MF_RESULT_NONE	Data being scanned is deleted
MF_RESULT_PARTIAL	Data being scanned can be acquired

## EndorsePrintMode

Set/Get the Print method for endorsement printing.

### Type

Property: Read/Write  
Data: enum MfEndorsePrintMode

### Values

enum MfEndorsePrintMode	Description
MF_ENDORSEPRINT_MODE_HIGHS PEED	If print data creation has not been completed within the period of time in which data transfer is permitted, the process goes on without printing.
MF_ENDORSEPRINT_MODE_DATAWAITING	If print data creation has not been completed within the period of time in which data transfer is permitted, waits at the print start position until data creation becomes complete.



If a Print Data Unreceived Error occurs, configuring this setting to MF\_ENDORSEPRINT\_MODE\_DATAWAITING is recommended.

## PrnDataLenExceedErrorEject

Set/Get the ejection method for when the printing content exceeds printable area.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket

## PrnDataLenExceedCancel

Set/Get the whether to continue reading operation when a paper length error occurs.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

## PrnDataUnreceiveErrorEject

Set/Get the ejection method for when the print data unreceived error is occurred.

### Type

Property: Read/Write  
Data: enum MfEject

### Values

enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket

## PrnDataUnreceiveCancel

Set/Get the whether to continue reading operation when a print data unreceived error occurs.

### Type

Property: Read/Write  
Data: enum MfCancel

### Values

enum MfCancel	Description
MF_CANCEL_DISABLE	Reading process for the next check sheet is canceled
MF_CANCEL_ENABLE	Reading process for the next check sheet is not canceled

# MFlqa Class - Properties

This sections contains descriptions of all of the properties exposed by the MFlqa class.

## Version

This is the class version.

### Type

Property: Read only

Data: int

### Values

The version of class used internally within this class.

## ErrorSelect

This sets whether to detect the image quality defects (IQA error).

### Type

Property: Read/Write

Data: Enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	No error detected
MF_ERROR_SELECT_DETECT	Error detected

## ErrorEject

This sets the ejection method for when IQA error is detected. The valid commands are listed below.

### Type

Property: Read/Write

Data: Enum MfEject

### Values

Enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## Stamp

This sets the whether to enable a franker for when the IQA error is detected.

### Type

Property: Read/Write  
Data: Enum MfStamp

### Values

Enum MfStamp	Description
MF_STAMP_ENABLE	Franker enabled
MF_STAMP_DISABLE	Franker disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## Cancel

Sets whether to continue reading operation when the IQA error occurs.

### Type

Property: Read/Write  
Data: Enum MfCancel

### Values

Enum MfCancel	Description
MF_CANCEL_DISABLE	Does not cancel the reading process for the next check sheet.
MF_CANCEL_ENABLE	Cancels the reading process for the next check sheet.

## ImageFormat

This sets an image format at the IQA validation.

### Type

Property: Read/Write  
Data: Enum Format

### Values

Format	Description
EPS_BI_SCN_TIFF	TIFF format CCITT (Group 4) compressed data
EPS_BI_SCN_RASTER	Raster format uncompressed data
EPS_BI_SCN_BITMAP	Bitmap format uncompressed data
EPS_BI_SCN_TIFF256	TIFF format uncompressed data
EPS_BI_SCN_JPEGHIGH	JPEG format high compression (size priority) data
EPS_BI_SCN_JPEGNORMAL	JPEG format normal compression data
EPS_BI_SCN_JPEGLOW	JPEG format low compression (quality priority) data
EPS_BI_SCN_JTIFF	TIFF format JPEG compressed data

## ColorDepth

This sets the gradation (bits per pixel) at the IQA validation.

### Type

Property: Read/Write  
Data: Enum ColorDepth

### Values

ColorDepth	Description
EPS_BI_SCN_1BIT	Black and White (1 bit)
EPS_BI_SCN_8BIT	256 grayscale (8 bit)

## Threshold

This sets the density threshold at the IQA validation.

### Type

Property: Read/Write  
Data: double

### Values

The valid value is -128 to 127.

## Color

This sets color at the IQA validation.

### Type

Property: Read/Write

Data: Enum Color

### Values

Color	Description
EPS_BI_SCN_MONOCHROME	Monochrome
EPS_BI_SCN_COLOR	Color

## ExOption

This sets the variety of density adjustment at the IQA validation.

### Type

Property: Read/Write

Data: Enum ExOption

### Values

ExOption	Description
EPS_BI_SCN_MANUAL	Applies the value of bThreshold for Black and White.
EPS_BI_SCN_SHARP	Sharpening
EPS_BI_SCN_SHARP_CUSTOM	Custom sharpening
EPS_BI_SCN_SHARP_CUSTOM2	Combination of sharpening and custom sharpening

## Resolution

This sets the resolution at the IQA validation.

### Type

Property: Read/Write  
Data: Enum MfScanDpi

### Values

sResolution	Description
MF_SCAN_DPI_DEFAULT	Scans at the default resolution for the device. (200 DPI)
MF_SCAN_DPI_200	Scans at 200 DPI.
MF_SCAN_DPI_300	Scans at 300 DPI.

Each of the following values can be specified independently only if EPS\_BI\_SCN\_UNIT\_CHECKPAPER is set to the unit when this class is used. (Otherwise, it is considered to be not defined.)

sResolution (Constant)	Description
MF_SCAN_DPI_100	Scans at 100 DPI.
MF_SCAN_DPI_120	Scans at 120 DPI.
MF_SCAN_DPI_240	Scans at 240 DPI.

The following value can be specified independently only if EPS\_BI\_SCN\_UNIT\_CARD is set to the unit when this class is used. (Otherwise, it is considered to be not defined.)

sResolution (Constant)	Description
MF_SCAN_DPI_600	Scans at 600 DPI.

## Undersize

This sets the execution of UndersizeImage validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Undersize	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Oversize

This sets the execution of OversizeImage validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Oversize	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Mincompressed

This sets the execution of MinCompressedImageSize validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Mincompressed	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Maxcompressed

This sets the execution of MaxCompressedImageSize validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

## Values

Maxcompressed	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Front\_rear

This sets the execution of FrontRearImageMismatch validation.

## Type

Property: Read/Write  
Data: Enum MflqaTest

## Values

Front_rear	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Toolight

This sets the execution of ImageTooLight validation.

## Type

Property: Read/Write  
Data: Enum MflqaTest

## Values

Toolight	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Toodark

This sets the execution of ImageTooDark validation.

## Type

Property: Read/Write  
Data: Enum MflqaTest

## Values

Toodark	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Streaks

This sets the execution of HorizontalStreaksPresent validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Streaks	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Noise

This sets the execution of ExcessiveSpotNoise validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Noise	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Focus

This sets the execution of ImageOutOfFocus validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Focus	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Corners

This sets the execution of FoldedTornDocCorners validation.

### Type

Property: Read/Write

Data: Enum MflqaTest

### Values

Corners	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Edges

This sets the execution of FoldedTornDocEdges validation.

### Type

Property: Read/Write

Data: Enum MflqaTest

### Values

Edges	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Framing

This sets the execution of DocFramingError validation.

### Type

Property: Read/Write

Data: Enum MflqaTest

### Values

Framing	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Skew

This sets the execution of ExcessiveDocSkew validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Skew	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Carbon

This sets the execution of ExcessiveDocSkew validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Skew	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

## Piggyback

This sets the execution of Piggyback validation.

### Type

Property: Read/Write  
Data: Enum MflqaTest

### Values

Piggyback	Description
MF_IQA_TEST_DISABLE	Validation is not executed.
MF_IQA_TEST_ENABLE	Validation is executed.

# MFlqaResult Class - Properties

This sections contains descriptions of all of the properties exposed by the MFlqaResult class.

## Version

This is the class version.

### Type

Property: Read only

Data: int

### Values

The version of class used internally within this class.

## Ret

This sets the status when the scan is completed.

### Type

Property: Read only

Data: Enum ErrorCode

### Values

Ret	Value	Description
SUCCESS	0	Success
ERR_SCAN	-450	Device failed in image scanning
ERR_SCN_IQA	-1120	Error is detected by the IQA validation

## UndersizeImage

UndersizeImage validation result is set.

### Type

Property: Read only  
Data: IqaResultUnderSizeImageClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

Width

Image width (unit: 0.1 inch)

Height

Image height (unit: 0.1 inch)

## OversizeImage

OversizeImage validation result is set.

### Type

Property: Read only  
Data: IqaResultOverSizeImageClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

Width

Image width (unit: 0.1 inch)

Height

Image height (unit: 0.1 inch)

## MaxCompressedImageSize

MaxCompressedImageSize validation result is set.

### Type

Property: Read only

Data: IqaResultMaxCompressedImageSizeClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

iSize

Image width (unit: Byte)

## MinCompressedImageSize

MinCompressedImageSize validation result is set.

### Type

Property: Read only

Data: IqaResultMinCompressedImageSizeClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

iSize

Compressed size (unit: Byte)

## FrontRearImageMismatch

FrontRearImageMismatch validation result is set.

### Type

Property: Read only  
Data: IqaResultFrontRearImageMismatchClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

Width

Width difference between the front-side and back-side images (unit: 0.1 inch)

Height

Height difference between the front-side and back-side images (unit: 0.1 inch)

## ImageTooLight

ImageTooLight validation result is set.

### Type

Property: Read only  
Data: IqaResultImageTooLightClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

BlackPixels

Percentage of black pixels in the image (unit: 0.1%)

Brightness

Image brightness (unit: 0.1%)

Contrast

Image contrast (unit: 0.1%)

## ImageTooDark

ImageTooDark validation result is set.

### Type

Property: Read only  
Data: IqaResultImageTooDarkClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

iBlackPixels

Percentage of black pixels in the image (unit: 0.1%)

Brightness

Image brightness (unit: 0.1%)

Contrast

Image contrast (unit: 0.1%)

## HorizontalStreaksPresent

HorizontalStreaksPresent validation result is set.

### Type

Property: Read only  
Data: IqaResultHorizontalStreaksPresentClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

StreakCount

Number of black stripes present in binary images

StreakHeight

Width of the thickest black stripe

## ExcessiveSpotNoise

ExcessiveSpotNoise validation result is set.

### Type

Property: Read only  
Data: IqaResultExcessiveSpotNoiseClass

### Values

#### Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

#### iCount

Mean number of spots (noise) per square inch

## ImageOutOfFocus

ImageOutOfFocus validation result is set.

### Type

Property: Read only  
Data: IqaResultImageOutOfFocusClass

### Values

#### Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

#### ImageFocusScore

Score estimated by (max video gradient) / (grey level dynamic range) \* (pixel pitch)

## FoldedTornDocCorners

FoldedTornDocCorners validation result is set.

### Type

Property: Read only

Data: IqaResultFoldedTornDocCornersClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

TopLeftWidth

Width of a tear/bend at the upper left corner of the image (unit: 0.1 inch)

TopLeftHeight

Height of a tear/bend at the upper left corner of the image (unit: 0.1 inch)

TopRightWidth

Width of a tear/bend at the upper right corner of the image (unit: 0.1 inch)

TopRightHeight

Height of a tear/bend at the upper right corner of the image (unit: 0.1 inch)

BottomLeftWidth

Width of a tear/bend at the lower left corner of the image (unit: 0.1 inch)

BottomLeftHeight

Height of a tear/bend at the lower left corner of the image (unit: 0.1 inch)

BottomRightWidth

Width of a tear/bend at the lower right corner of the image (unit: 0.1 inch)

BottomRightHeight

Height of a tear/bend at the lower right corner of the image (unit: 0.1 inch)

## FoldedTornDocEdges

FoldedTornDocEdges validation result is set.

### Type

Property: Read only  
Data: IqaResultFoldedTornDocEdgesClass

### Values

#### Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

#### TopLeftWidth

Width of a tear/bend at the upper left corner of the image (unit: 0.1 inch)

#### TopLeftHeight

Height of a tear/bend at the upper left corner of the image (unit: 0.1 inch)

#### TopRightWidth

Width of a tear/bend at the upper right corner of the image (unit: 0.1 inch)

#### TopRightHeight

Height of a tear/bend at the upper right corner of the image (unit: 0.1 inch)

#### BottomLeftWidth

Width of a tear/bend at the lower left corner of the image (unit: 0.1 inch)

#### BottomLeftHeight

Height of a tear/bend at the lower left corner of the image (unit: 0.1 inch)

#### BottomRightWidth

Width of a tear/bend at the lower right corner of the image (unit: 0.1 inch)

#### BottomRightHeight

Height of a tear/bend at the lower right corner of the image (unit: 0.1 inch)

## DocFramingError

DocFramingError validation result is set.

### Type

Property: Read only  
Data: IqaResultDocFramingErrorClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

Top

Width of the top margin of the image (unit: 0.1 inch)

Left

Width of the left margin of the image (unit: 0.1 inch)

Right

Width of the right margin of the image (unit: 0.1 inch)

Bottom

Width of the bottom margin of the image (unit: 0.1 inch)

## ExcessiveDocSkew

ExcessiveDocSkew validation result is set.

### Type

Property: Read only  
Data: IqaResultExcessiveDocSkewClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

iAngle

Tilt angle (unit: 0.1 degree)

iRange

Fixed to 0: Value indicating that iAngle is in the range from -90 to +90

## CarbonStripDetection

CarbonStripDetection validation result is set.

### Type

Property: Read only  
Data: IqaResultCarbonStripDetectionClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

iStripHeight

Length of a carbon strip (unit: 0.1 inch)

## Piggyback

Piggyback validation result is set.

### Type

Property: Read only  
Data: IqaResultPiggybackClass

### Values

Result

Result	Description
IQARESULT_NOT_TESTED	The test has not been performed.
IQARESULT_PASS	The test was passed.
IQARESULT_NOT_PASS	The test was failed.

# MFBarcode Class - Properties

This sections contains descriptions of all of the properties exposed by the MFBarcode class.

## Version

This is the class version.

### Type

Property: Read only  
Data: int

### Values

The version of class used internally within this class.

## Ret

Obtains the execution result of barcodes.

### Type

Property: Read only  
Data: Enum ErrorCode

### Values

Refer to [MFBarcode.Ret \(p.77\)](#).

## ErrorSelect

Specifies whether to detect the barcode decode error.

### Type

Property: Read/Write  
Data: Enum MfErrorSelect

### Values

enum MfErrorSelect	Description
MF_ERROR_SELECT_NODETECT	Not detect an external noise error.
MF_ERROR_SELECT_DETECT	Detect an external noise error.

## ErrorEject

Specifies where to eject when the barcode decode error is detected. The valid commands are listed below.

### Type

Property: Read/Write  
Data: Enum MfEject

### Values

Enum MfEject	Description
MF_EJECT_MAIN_POCKET	Ejected to the main pocket
MF_EJECT_SUB_POCKET	Ejected to the sub pocket
MF_EJECT_NOEJECT	Not ejected (completed with error)

## Stamp

Specifies whether to do the franking process to the sheet when the barcode decode error is detected. The valid commands are listed below.

### Type

Property: Read/Write  
Data: Enum MfStamp

### Values

bStamp (Constant)	Description
MF_STAMP_ENABLE	Franker enabled
MF_STAMP_DISABLE	Franker disabled



For TM-S9000/S2000 API, regardless of this value, the action is always not to stamp.

## Cancel

Specifies whether to continue the reading process when the barcode decode error is detected. The valid commands are listed below.

### Type

Property: Read/Write  
Data: Enum MfCancel

### Values

Enum MfCancel	Description
MF_CANCEL_DISABLE	Does not cancel the reading process for the next check sheet.
MF_CANCEL_ENABLE	Cancels the reading process for the next check sheet.

## TargetColor

Specifies the target color for barcode decoding.

### Type

Property: Read/Write  
Data: Enum MfBarcodeTargetColor

### Values

enum MfErrorSelect	Description
MF_BARCODE_TARGET_COLOR_GRAY	Decode as grayscale

## Resolution

Specifies the image resolution of image data. The valid commands are listed below.

### Type

Property: Read/Write  
Data: Enum MfScanDpi

### Values

Enum MfScanDpi	Description
MF_SCAN_DPI_DEFAULT	Scans at the default resolution for the device. (200 DPI)
MF_SCAN_DPI_200	Scans at 200 DPI.
MF_SCAN_DPI_300	Scans at 300 DPI.

Each of the following values can be specified independently only if EPS\_BI\_SCN\_UNIT\_CHECKPAPER is set to the unit when this class is used. (Otherwise, it is considered to be not defined.)

sResolution (Constant)	Description
MF_SCAN_DPI_100	Scans at 100 DPI.
MF_SCAN_DPI_120	Scans at 120 DPI.
MF_SCAN_DPI_240	Scans at 240 DPI.

The following value can be specified independently only if EPS\_BI\_SCN\_UNIT\_CARD is set to the unit when this class is used. (Otherwise, it is considered to be not defined.)

sResolution (Constant)	Description
MF_SCAN_DPI_600	Scans at 600 DPI.

## InfoMode

Specify the stInfo sequence.

### Type

Property: Read/Write  
Data: int

### Values

int	Description
0	Regions are not specified according to information. If 0 is specified, the only stInfo(0) elements referred to are SymbolMask and Direction.
1 to 5	Specification of regions according to information is enabled. By specifying in InfoMode the number of items in the MFBarcode.Barcodeinfo class to enable, it is possible to make multiple settings related to decoding. Example: If 3 is specified for InfoMode, Info(0), Info(1), and Info(2) are referenced, and Info(3) and Info(4) are ignored.

## Info

This is the class where the decode setting and the status of decode result are made. Five arrangements are available. Refer to [MFBarcode.Barcodeinfo Class - Properties \(p.257\)](#).

### Type

Property:	Read/Write
Data:	MFBarcode.BarcodeInfo(5)

## Data

The result of barcode decoding (MFBarcode.BarcodeData class) is set.

The MFBarcode.BarcodeData class is an element of arrangement. If there is no decode result of barcode, null is set. Refer to [MFBarcode.BarcodeData Class - Properties \(p.261\)](#).

### Type

Property:	Read only
Data:	MFBarcode.BarcodeData()

## MFBarcode.Barcodeinfo Class - Properties

This sections contains descriptions of all of the properties exposed by the MFBarcode.Barcodeinfo class.

### Ret

The barcode decode execution result is set.

#### Type

Property: Read only  
Data: Enum ErrorCode

#### Values

Refer to [MFBarcode.Ret \(p.77\)](#) for details.

### Status

The status of decoding result is set.

#### Type

Property: Read only  
Data: byte

#### Values

Bit	Status Contents	ON / OFF	Value	Status
0	---	---	0x0000	Reserved (Fixed to 0)
1	---	---	0x0000	Reserved (Fixed to 0)
2	---	---	0x0000	Reserved (Fixed to 0)
3	Detailed information	ON	0x0008	Added
		OFF	0x0000	-
4	---	---	0x0000	Reserved (Fixed to 0)
5	Reading result	ON	0x0020	Failure end
		OFF	0x0000	Success
6	---	---	0x0000	Reserved (Fixed to 0)
7	---	---	0x0000	Reserved (Fixed to 0)

## Detail

The detailed status of decoding result is set.

### Type

Property: Read only

Data: byte

### Values

Value	Information
40h	Success
45h	Barcode cannot be detected.

## SymbolMask

Set/Get the type of barcode symbol to decode. Make sure to set this because the default value is not set for this setting.

### Type

Property: Read/Write

Data: Enum MfBarcodeSymbol

### Values

Enum BarcodeSymbol	Barcode Type
MF_BARCODE_SYMBOL_CODABAR	Codabar
MF_BARCODE_SYMBOL_CODE128	Code128
MF_BARCODE_SYMBOL_CODE39	Code39
MF_BARCODE_SYMBOL_ITF	ITF
MF_BARCODE_SYMBOL_EAN_JAN	JAN13(EAN), JAN8(EAN)
MF_BARCODE_SYMBOL_UPC_A	UPC-A
MF_BARCODE_SYMBOL_UPC_E	UPC-E

## Direction

Set/Get the barcode decode direction.

### Type

Property: Read/Write  
Data: Enum MfBarcodeDirection

### Values

Enum Direction (Constant)	Description
MF_BARCODE_DIRECTION_ALL	Decodes it in both directions from left to right and top to bottom
MF_BARCODE_DIRECTION_LEFTRIGHT	Decodes it from left to right
MF_BARCODE_DIRECTION_TOPBOTTOM	Decodes it from top down
MF_BARCODE_DIRECTION_RIGHTLEFT	Decodes it from right to left
MF_BARCODE_DIRECTION_BOTTOMTOP	Decodes it from bottom up

## Origin

Set/Get the origin to specify the decode area.

### Type

Property: Read/Write  
Data: Enum MfBarcodeOrigin

### Values

Enum MfBarcodeOrigin	Description
MF_BARCODE_ORIGIN_TOP_LEFT	Sets the origin to the top left corner in relation to the document insertion direction.
MF_BARCODE_ORIGIN_BOTTOM_LEFT	Sets the origin to the bottom left corner in relation to the document insertion point.
MF_BARCODE_ORIGIN_TOP_RIGHT	Sets the origin to the top right corner in relation to the document insertion point.
MF_BARCODE_ORIGIN_BOTTOM_RIGHT	Sets the origin to the bottom right corner in relation to the document insertion point.

## StartX

Set/Get the x-coordinate (unit: 0.1mm) of started decode.

### Type

Property:	Read/Write
Data:	int

## StartY

Set/Get they-coordinate (unit: 0.1mm) of started decode.

### Type

Property:	Read/Write
Data:	int

## EndX

Set/Get the x-coordinate of ended decode.

### Type

Property:	Read/Write
Data:	int

### Values

Larger than StartX (unit: 0.1mm)

## EndY

Set/Get they-coordinate of ended decode.

### Type

Property:	Read/Write
Data:	int

### Values

Larger than StartY (unit: 0.1mm)

## MFBarcode.BarcodeData Class - Properties

This sections contains descriptions of all of the properties exposed by the MFBarcode.BarcodeData class.

### Symbol

Get the scanned barcode symbol.

#### Type

Property: Read only  
Data: Enum MfBarcodeSymbol

#### Values

Enum Symbol	Barcode Type
MF_BARCODE_SYMBOL_CODABAR	Codabar
MF_BARCODE_SYMBOL_CODE128	Code128
MF_BARCODE_SYMBOL_CODE39	Code39
MF_BARCODE_SYMBOL_ITF	ITF
MF_BARCODE_SYMBOL_EAN_JAN	JAN13(EAN), JAN8(EAN)
MF_BARCODE_SYMBOL_UPC_A	UPC-A
MF_BARCODE_SYMBOL_UPC_E	UPC-E

### Direction

Get the barcode decode direction.

#### Type

Property: Read only  
Data: Enum MfBarcodeDirection

#### Values

Enum Direction (Constant)	Description
MF_BARCODE_DIRECTION_ALL	Decodes it in both directions from left to right and top to bottom
MF_BARCODE_DIRECTION_LEFTRIGHT	Decodes it from left to right
MF_BARCODE_DIRECTION_TOPBOTTOM	Decodes it from top down
MF_BARCODE_DIRECTION_RIGHTLEFT	Decodes it from right to left
MF_BARCODE_DIRECTION_BOTTOMTOP	Decodes it from bottom up

## StartX

Get the x-coordinate of started decode.

### Type

Property:	Read only
Data:	int

## StartY

Get the y-coordinate of started decode.

### Type

Property:	Read only
Data:	int

## EndX

Get the x-coordinate of ended decode.

### Type

Property:	Read only
Data:	int

## EndY

Get the y-coordinate of ended decode.

### Type

Property:	Read only
Data:	int

## Data

The scanned barcode's binary data is set.

The size including the null at the end is set if the set result is a character string.

Free the reference memory with GlobalFree.

### Type

Property:	Read only
Data:	byte[]

## MFOcrAb Class - Properties

This sections contains descriptions of all of the properties exposed by the MFOcrAb class.

### Version

This is the class version.

#### Type

Property: Read only

Data: int

#### Values

The version of class used internally within this class.

### Ret

The return value for this function is set.

#### Type

Property: Read only

Data: Enum ErrorCode

#### Values

ErrorCode	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Insufficient memory
ERR_HANDLE	-60	Invalid handle value
ERR_PARAM	-90	Parameter error
ERR_NOT_FOUND	-220	Data not found
ERR_IMAGE_FILEOPEN	-230	Open failure
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_MICR_NODATA	-1040	There is no MICR data. Determine whether to scan again.

## OcrType

### Type

Property: Read/Write  
Data: Enum MfOcrType

### Values

Enum MfOcrType	Description
MF_OCR_FONT_OCRA_NUM	Numeric characters in OCR-A font
MF_OCR_FONT_OCRB_NUM	Numeric characters in OCR-B font
MF_OCR_FONT_OCRA_ALPHA	Alphabetic characters in OCR-A font
MF_OCR_FONT_OCRB_ALPHA	Alphabetic characters in OCR-B font
MF_OCR_FONT_OCRA_ALPHANUM	Alphanumeric characters in OCR-A font
MF_OCR_FONT_OCRB_ALPHANUM	Alphanumeric characters in OCR-B font
MF_OCR_FONT_OCRA_ALPHANUM_WOOH	Alphanumeric characters in OCR-A font (except OH)
MF_OCR_FONT_OCRB_ALPHANUM_WOOH	Alphanumeric characters in OCR-B font (except OH)
MF_OCR_FONT_OCRA_ALPHANUM_WOZERO	Alphanumeric characters in OCR-A font (except ZERO)
MF_OCR_FONT_OCRB_ALPHANUM_WOZERO	Alphanumeric characters in OCR-B font (except ZERO)
MF_OCR_FONT_OCRA_SYMNUM	OCR-A font, numeric numbers and symbols ("+" excluded)
MF_OCR_FONT_OCRB_SYMNUM	OCR-B font, numeric numbers and symbols ("+" included)

## Direction

### Type

Property: Read/Write  
Data: Enum MfDirection

### Values

Enum MfDirection	Description
MF_OCR_LEFTRIGHT	From left to right (normal direction)
MF_OCR_TOPBOTTOM	From top to bottom (rotated 90 degrees clockwise)
MF_OCR_RIGHTLEFT	From right to left (upside-down)
MF_OCR_BOTTOMTOP	From bottom to top (rotated 90 degrees counterclockwise)

## StartX

Specify start X coordinate of the OCR recognition area.

### Type

Property: Read/Write  
Data: uShort

### Values

The allowable range is between 0 and 254 (unit: mm).

A parameter error will be returned if a value outside the range is specified.

## StartY

Specify start Y coordinate of the OCR recognition area.

### Type

Property: Read/Write  
Data: uShort

### Values

The allowable range is between 0 and 255 (unit: mm).

A parameter error will be returned if a value outside the range is specified.

## EndX

Specify end X coordinate of the OCR recognition area.

### Type

Property: Read/Write  
Data: uShort

### Values

The settable values are 1 through 255, OCR\_AREA\_RIGHT, and OCR\_AREA\_LEFT (unit: mm).

When OCR\_AREA\_RIGHT is specified, the right side of an image can be specified.

When OCR\_AREA\_LEFT is specified, the left side of an image can be specified.

In the following cases, the parameter error (ERR\_PARAM) is returned:

- When a value other than OCR\_AREA\_RIGHT or OCR\_AREA\_LEFT is specified in the specified reading range and StartX  $\geq$  EndX is set
- When Direction is set to MF\_OCR\_TOPBOTTOM or MF\_OCR\_BOTTOMTOP

## EndY

Specify end Y coordinate of the OCR recognition area.

### Type

Property: Read/Write  
Data: uShort

### Values

The settable values are 1 through 256, OCR\_AREA\_BOTTOM, and OCR\_AREA\_TOP (unit: mm).

If OCR\_AREA\_BOTTOM is specified when OCR origin is set to top left or top right, the bottom of the image can be specified.

If OCR\_AREA\_TOP is specified when OCR origin is set to bottom left or bottom right, the top of the image can be specified.

In the following cases, the parameter error (ERR\_PARAM) is returned:

- when a value other than OCR\_AREA\_BOTTOM or OCR\_AREA\_TOP is specified in the specified reading range and StartY >= EndY is set
- when Direction is set to MF\_OCR\_LEFTRIGHT or MF\_OCR\_RIGHTLEFT

## SpaceHandling

### Type

Property: Read/Write  
Data: Enum MfSpaceHandling

### Values

Enum MfSpaceHandling	Description
OCR_SPACE_ENABLE	Include space characters in OCR recognition result
OCR_SPACE_DISABLE	Not include space characters in OCR recognition result

## OcrStr

Store character strings acquired by OCR recognition process.

### Type

Property: Read only  
Data: String

## OcrReliableInfo.FirstSelectString

Best guess raw data read optically from OCR characters.

### Type

Property:	Read only
Data:	String

## OcrReliableInfo.SecondSelectString

Second guess raw data read optically from OCR characters.

### Type

Property:	Read only
Data:	String

## OcrReliableInfo.FirstSelectPercentage

Confidence level of best guess raw data read optically from OCR characters.

### Type

Property:	Read only
Data:	int()

## OcrReliableInfo.SecondSelectPercentage

Confidence level of second guess raw data read optically from OCR characters.

### Type

Property:	Read only
Data:	int()

# MFVersion Class - Properties

This sections contains descriptions of all of the properties exposed by the MFVersion class.

## Description

Acquires the information of the driver name.

### Type

Property:	Read only
Data:	String

## Version

Acquires the version information.

### Type

Property:	Read only
Data:	String

## MFPrintAreaInfo Class - Properties

This sections contains descriptions of all of the properties exposed by the MFPrintAreaInfo class.

### AreaName

Sets/Acquires the area name.

#### Type

Property:	Read/Write
Data:	String

### OriginX

Sets/Acquires The x coordinate of the origin position. The top left of the printable area is set as the origin. If print starting position is changed to a setting based on the back-end of the paper using SetConfigure API, the top right of the printable area is set as the origin.

#### Type

Property:	Read/Write
Data:	int

#### Values

Maximum value that can be designated for the X direction larger than 0.

### OriginY

Sets/Acquires the y coordinate of the origin position. The top left of the printable area is set as the origin.

#### Type

Property:	Read/Write
Data:	int

#### Values

Maximum value that can be designated for the Y direction larger than 0.

### Width

Sets/Acquires the area width.

#### Type

Property:	Read/Write
Data:	int

#### Values

Maximum value that can be designated for the X direction larger than 1.

## Height

Sets/Acquires the area height.

### Type

Property: Read/Write  
Data: int

### Values

Maximum value that can be designated for the Y direction larger than 1.

## Rotate

Sets/Acquires the coordinates of the origin, and the unit of measurement for the values specified for the area width and area height.

### Type

Property: Read/Write  
Data: enum ImageRotate

### Values

enum ImageRotate	Description
MEASURE_MM	The unit of measurement should be mm.
MEASURE_INCH_01	The unit of measurement should be 0.1 inches.

## Measure

Sets/Acquires the area rotation.

### Type

Property: Read/Write  
Data: enum ImageMeasure

### Values

enum ImageMeasure	Description
IMAGEROTATE_0	The printing does not rotate.
IMAGEROTATE_90	The printing rotates 90 degrees counterclockwise around the origin.
IMAGEROTATE_180	The printing rotates 180 degrees counterclockwise around the origin.
IMAGEROTATE_270	The printing rotates 270 degrees counterclockwise around the origin.

## MFPrint Class - Properties

This sections contains descriptions of all of the properties exposed by the MFPrint class.



This class is used only for compatibility with the TM-J9000/TM-S1000 drivers. On this product, please print using the printing-related API (such as PrintText).

### Version

This is the class version.

#### Type

Property: Read only  
Data: int

#### Values

The version of class used internally within this class.

### Ret

The return value for this function is set.

#### Type

Property: Read only  
Data: Enum ErrorCode

#### Values

Refer to [MFBarcode.Ret \(p.77\)](#).

### EndorsementType

Sets/Acquires the transaction printing process and the electronic endorsement process.

#### Type

Property: Read/Write  
Data: Enum EndorsementType

#### Values

EndorsementType	Description
MF_PRINT_TYPE_ELECTRIC_ENDORSE_ONLY	Electronic endorsement only
MF_PRINT_TYPE_ELECTRIC_ENDORSE_EXTEND	Electronic endorsement only
MF_PRINT_TYPE_ENDORSE_NORMAL	Performs both endorsement printing and electronic endorsement printing by using data specified by String.
MF_PRINT_TYPE_ENDORSE_EXTEND	Executes endorsement printing using the data specified with String.

## String[3]

Sets/Acquires the ASCII character string for electronic endorsement printing.

### Type

Property: Read/Write  
Data: String

### Values

Sets/Acquires the transaction printing process and the electronic endorsement process.

## Attribute[3]

Sets/Acquires the attribute of the character string. The attribute is specified with the following bits and multiple attributes can be specified.

### Type

Property: Read/Write  
Data: int

### Values

Attribute	Description
MF_PRINT_BOLD	Executes emphasized printing
MF_PRINT_UNDERLINE_1	Adds a 1-line width of UnderLine
MF_PRINT_UNDERLINE_2	Adds a two-line width of UnderLine
MF_PRINT_REVERSEVIDEO	Executes reverse printing
MF_PRINT_BLACK	Prints a character with the first color (usually it is black)
MF_PRINT_COLOR	<ul style="list-style-type: none"><li>Prints a character with the second color.</li><li>Even if this value is set, text is printed using the 1st color.</li></ul>
MF_PRINT_MIXED	<ul style="list-style-type: none"><li>Prints a character with the first and second colors</li><li>Even if this value is set, text is printed using the 1st color.</li></ul>
MF_PRINT_NO_ATTRIBUTE	Not specifying the attribute

## Font[3]

Sets/Acquires the text font.

### Type

Property: Read/Write  
Data: Enum FontType

### Values

Enum FontType	Value	Description
MF_PRINT_FONT_A	1	Device internal font A
MF_PRINT_FONT_B	2	Device internal font B

## Speed

Not used.

## Direction

Not used.

# Device Information

## Device Status

\* Not supported in TM-S2000.

Macro Definition (Constant)	ON / OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	Device not responding
	OFF	0x00000000	Device responding
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK *	ON	0x00000004	Status of drawer kick connector 3rd pin became high
	OFF	0x00000000	-
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_MAIN_POCKET_NEAR_FULL	ON	0x00000010	The main pocket is nearly full
	OFF	0x00000000	The main pocket is not nearly full
ASB_COVER_OPEN	ON	0x00000020	Cover is open
	OFF	0x00000000	Cover is closed
ASB_PAPER_FEED *	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_SUB_POCKET_NEAR_FULL	ON	0x00000080	The sub pocket is nearly full
	OFF	0x00000000	The sub pocket is not nearly full
ASB_WAIT_PEPRT_EJECT	ON	0x00000100	Waiting for cut sheet ejection
	OFF	0x00000000	-
ASB_PANEL_SWITCH *	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Recoverable errors generated
	OFF	0x00000000	Recoverable errors not generated
ASB_AUTOCUTTER_ERR *	ON	0x00000800	Auto-cutter error occurred
	OFF	0x00000000	Auto-cutter error not occurred
ASB_UNRECOVER_ERR	ON	0x00002000	Unrecoverable error occurred
	OFF	0x00000000	Unrecoverable error not occurred
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto-recovery error occurred
	OFF	0x00000000	Auto-recovery error not occurred
ASB_NOT_CARD_INSERT	ON	0x00010000	The card is not inserted into the PhotoID detector
	OFF	0x00000000	-
ASB_RECEIPT_NEAR_END *	ON	0x00020000	No paper in the roll paper near end detector
	OFF	0x00000000	Paper in the roll paper near end detector
ASB_EJECT_SENSOR_NO_PAPER	ON	0x00040000	No paper in the eject detector
	OFF	0x00000000	Paper in the eject detector
ASB_RECEIPT_END *	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector

\* Not supported in TM-S2000.

Macro Definition (Constant)	ON / OFF	Value	Status
ASB_PAPER_INTERMEDIATE	ON	0x00200000	No paper in the intermediate detector
	OFF	0x00000000	Paper in the intermediate detector
ASB_ASF_PAPER	ON	0x00400000	No paper in the ASF detector
	OFF	0x00000000	Paper in the ASF detector
ASB_SLIP_SELECTED	ON	0x01000000	Cut sheet is not selected
	OFF	0x00000000	Cut sheet is selected
ASB_PRINT_SLIP	ON	0x02000000	Printing is disabled for cut sheet
	OFF	0x00000000	Printing is enabled for cut sheet
ASB_VALIDATION_SELECTED	ON	0x04000000	Validation is not selected
	OFF	0x00000000	Validation is selected
ASB_PRINT_VALIDATION	ON	0x08000000	Printing is disabled for validation
	OFF	0x00000000	Printing is enabled for validation
ASB_WAIT_INSERT	ON	0x20000000	Waiting for cut sheet insertion
	OFF	0x00000000	-
ASB_SLIP_PAPER_SIZE	ON	0x40000000	No paper in the paper length detector
	OFF	0x00000000	Paper in the paper length detector
ASB_VALIDATION_NO_PAPER (TM-J9000 Compatibility)	ON	0x40000000	No paper in the validation detector
	OFF	0x00000000	Paper in the validation detector

**Device Status (for TM-S1000)**

Macro Definition (Constant)	ON / OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	Device not responding
	OFF	0x00000000	Device responding
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_COVER_OPEN	ON	0x00000020	Cover is open
	OFF	0x00000000	Cover is closed
ASB_WAIT_PEPRT_EJECT	ON	0x00000100	Waiting for cut sheet ejection
	OFF	0x00000000	-
ASB_MECHANICAL_ERR	ON	0x00000400	Recoverable errors generated
	OFF	0x00000000	Recoverable errors not generated
ASB_UNRECOVER_ERR	ON	0x00002000	Unrecoverable error occurred
	OFF	0x00000000	Unrecoverable error not occurred
ASB_PAPER_INTERMEDIATE	ON	0x00010000	No paper in the intermediate detector
	OFF	0x00000000	Paper in the intermediate detector
ASB_MAIN_NEAR_FULL	ON	0x00020000	The main pocket is not nearly full
	OFF	0x00000000	The main pocket is nearly full
ASB_EJECT_SENSOR_NO_PAPER	ON	0x00040000	No paper in the eject detector
	OFF	0x00000000	Paper in the eject detector
ASB_SUB_NEAR_FULL	ON	0x00080000	The sub pocket is not nearly full
	OFF	0x00000000	The sub pocket is nearly full
ASB_SLIP_PAPER_SIZE	ON	0x00200000	No paper in the paper length detector
	OFF	0x00000000	Paper in the paper length detector
ASB_ASF_PAPER	ON	0x00400000	No paper in the ASF detector
	OFF	0x00000000	Paper in the ASF detector
ASB_STAMP_EXIST	ON	0x02000000	The franking cartridge is not installed
	OFF	0x00000000	The franking cartridge is installed
ASB_WAIT_INSERT	ON	0x20000000	Waiting for cut sheet insertion
	OFF	0x00000000	-
ASB_FRANKING_SENSOR	ON	0x40000000	No paper in the franking detector
	OFF	0x00000000	Paper in the franking detector

## Ink Status

Macro Definition (Constant)	ON / OFF	Value	Status
INK_ASB_OK	ON	0x0000	
INK_ASB_NEAR_END	ON	0x0001	Remaining ink level is low
	OFF	0x0000	-
INK_ASB_END	ON	0x0002	Exchange the ink cartridge
	OFF	0x0000	-
INK_ASB_NO_CARTRIDGE	ON	0x0004	No ink cartridge
	OFF	0x0000	-
INK_ASB_NO_CARTRIDGE2 *	ON	0x0008	No ink cartridge (2nd colors)
	OFF	0x0000	Ink cartridge present
INK_ASB_CLEANING	ON	0x0020	Cleaning in progress
	OFF	0x0000	Cleaning is not in progress
INK_ASB_NEAR_END2 *	ON	0x0100	Remaining ink level is low (2nd colors)
	OFF	0x0000	-
INK_ASB_END2 *	ON	0x0200	Exchange the ink cartridge (2nd colors)
	OFF	0x0000	-

\*: Defined for compatibility with TM-J9000. Normally OFF for this product.

## Maintenance Counter

\* TM-S9000 only

Resetability	Counter Number	Counter	Unit
Resetable	21 (15H) *	Count of thermal head energization	Count
	22 (16H) *	Number of lines fed by thermal head	Lines
	30 (1EH) *	Number of lines fed for roll paper	Lines
	31 (1FH)	Number of IJ head shots (Column A)	1000 shots
	32 (20H)	Number of IJ head shots (Column B)	1000 shots
	34 (22H)	Count of pump motor operations	Count
	50 (32H) *	Count of autocutter drive	Count
	60 (3CH)	Count of magnetic ink character read	Count
	61 (3DH)	Count of check paper scanning	Count
	62 (3EH)	Count of card scanning	Count
	67 (43H)	Count of check paper feeding	Count
	70 (46H)	Duration of product operation	Hours
	80 (50H)	Count of hopper open/close	Count
	82 (52H)	Count of pocket switch	Count
Cumulative	149 (95H) *	Count of thermal head energization	Count
	150 (96H) *	Number of lines fed by thermal head	Lines
	158 (9EH) *	Number of lines fed for roll paper	Lines
	159 (9EH)	Number of IJ head shots (Column A)	1000 shots
	160 (A0H)	Number of IJ head shots (Column B)	1000 shots
	162 (A2H)	Count of pump motor operations	Count
	178 (B2H) *	Count of autocutter drive	Count
	188 (BCH)	Count of magnetic ink character read	Count
	189 (BDH)	Count of check paper scanning	Count
	190 (BEH)	Count of card scanning	Count
	195 (C3H)	Count of check paper feeding	Count
	198 (C6H)	Duration of product operation	Hours
	208 (D0H)	Count of hopper open/close	Count
	210 (D2H)	Count of pocket switch	Count

## MICR Status

Bit	ON/OFF	Value	Status
0	-	0x00	Fixed to 0
1	-	0x02	Fixed to 1
2	ON	0x04	MICR function non-selection
	OFF	0x00	MICR function selection
3	ON	0x08	Waiting for check sheet/cleaning sheet insertion
	OFF	0x00	-
4	-	0x10	Fixed to 1
5	ON	0x20	Middle sensor senses no paper
	OFF	0x00	Middle sensor senses that there is paper
6	ON	0x40	ASF sensor senses no paper
	OFF	0x00	ASF sensor senses that there is paper
7	-	0x00	Fixed to 0

## Offline Code

### *GetOfflineCodeByIndex (for TM-S9000/S2000)*

First Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	-	0x00	Fixed to 0
1	-	0x00	Fixed to 0
2	-	0x00	Fixed to 0
3	-	0x00	Fixed to 0
4	OFF	0x00	Power retrigger error not generated (Fixed to 0)
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

Second Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	High voltage error generated
	OFF	0x00	High voltage error not generated
1	ON	0x02	Low voltage error generated
	OFF	0x00	Low voltage error not generated
2	-	-	Undefined
3	ON	0x08	FPGA CONFIG error generated
	OFF	0x00	FPGA CONFIG error not generated
4	ON	0x10	Ink-jet mechanical error generated
	OFF	0x00	Ink-jet mechanical error not generated
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Third Byte (Unrecovery Error)

\* Undefined in TM-S2000

Bit	ON/ OFF	Value	Status
0*	ON	0x01	Thermal head thermistor error generated
	OFF	0x00	Thermal head thermistor error not generated
1	ON	0x02	Ink-jet head thermistor error generated
	OFF	0x00	Ink-jet head thermistor error not generated
2	ON	0x04	Waste ink absorption volume error or ink information writing error generated
	OFF	0x00	Waste ink absorption volume error or ink information writing error not generated
3	ON	0x08	Motor driver thermistor error generated
	OFF	0x00	Motor driver thermistor error not generated
4	ON	0x10	Ink-jet head driver thermistor error generated
	OFF	0x00	Ink-jet head driver thermistor error not generated
5	ON	0x20	Pump drive frequency error generated
	OFF	0x00	Pump drive frequency error not generated
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Fourth Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	-	-	Undefined
1	-	-	Undefined
2	-	-	Undefined
3	-	-	Undefined
4	-	-	Undefined
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

Fifth Byte (Recoverable Error: Mechanism position error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Hopper position error generated
	OFF	0x00	Hopper position error not generated
1	ON	0x02	Switching plate position error generated
	OFF	0x00	Switching plate position error not generated
2	-	-	Undefined
3	-	-	Undefined
4	-	-	Undefined
5	ON	0x20	Endorsement printing data not received error generated
	OFF	0x00	Endorsement printing data not received error not generated
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

Sixth Byte (Recoverable Error: Roll paper)

\* Undefined in TM-S2000

Bit	ON/ OFF	Value	Status
0 *	ON	0x01	Auto-cutter error generated
	OFF	0x00	Auto-cutter error not generated
1 *	ON	0x02	Roll paper cover open error generated (when specifying recovery error)
	OFF	0x00	Roll paper cover open error not generated (when specifying recovery error)
2	-	-	Undefined
3	-	-	Undefined
4	-	-	Undefined
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Seventh Byte (Recoverable Error: Cut sheet paper)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Cut sheet paper stuck error generated
	OFF	0x00	Cut sheet paper stuck error not generated
1	ON	0x02	Cut sheet paper feed error generated
	OFF	0x00	Cut sheet paper feed error not generated
2	ON	0x04	Cut sheet paper length error generated
	OFF	0x00	Cut sheet paper length error not generated
3	ON	0x08	Cut sheet paper transfer error generated
	OFF	0x00	Cut sheet paper transfer error not generated
4	ON	0x10	Paper path cover open error generated, when cut sheet paper is printed
	OFF	0x00	Paper path cover open error not generated, when cut sheet paper is printed
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Eighth Byte (Recoverable Error: Other)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Cut sheet paper double feed error generated
	OFF	0x00	Cut sheet paper double feed error not generated
1	ON	0x02	Cut sheet paper insertion direction error generated
	OFF	0x00	Cut sheet paper insertion direction error not generated
2	ON	0x04	External noise error generated
	OFF	0x00	External noise error not generated
3	ON	0x08	Error in the read processing that was designated by a command
	OFF	0x00	No error in the read processing that was designated by a command
4	ON	0x10	Print specification error caused by specifying the printing content exceeds printable area
	OFF	0x00	Print specification error not caused by specifying the printing content exceeds printable area
5	ON	0x20	No magnetically read characters error generated
	OFF	0x00	No magnetically read characters error not generated
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Ninth Byte (Auto-Recovery Error)

\* Undefined in TM-S2000

Bit	ON/ OFF	Value	Status
0 *	ON	0x01	Roll paper cover open error generated (when specifying auto-recovery error)
	OFF	0x00	Roll paper cover open error not generated (when specifying auto-recovery error)
1	ON	0x02	Thermal head/ink-jet head driver temporary high-temperature error generated
	OFF	0x00	Thermal head/ink-jet head driver temporary high-temperature error not generated
2 *	ON	0x04	Motor driver IC temporary high-temperature error generated
	OFF	0x00	Motor driver IC temporary high-temperature error not generated
3	ON	0x08	Ink carriage cover is open
	OFF	0x00	Ink carriage cover is closed
4	ON	0x10	Paper path cover is open
	OFF	0x00	Paper path cover is closed
5 *	ON	0x20	Roll paper cover is open
	OFF	0x00	Roll paper cover is closed
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Tenth Byte (Auto-Recovery Error)

\* Undefined in TM-S2000

Bit	ON/ OFF	Value	Status
0 *	ON	0x01	Thermal head temporary low-voltage error generated
	OFF	0x00	Thermal head temporary low-voltage error not generated
1	-	0x00	Fixed to 0
2	-	0x00	Fixed to 0
3	-	0x00	Fixed to 0
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

**GetOfflineCodeByIndex (for TM-S1000)**

## First Byte

Bit	ON/ OFF	Value	Status
0	-	0x00	Fixed to 0
1	ON	0x02	CPU execution error generated
	OFF	0x00	CPU execution error not generated
2	ON	0x04	ROM error generated
	OFF	0x00	ROM error not generated
3	-	0x00	Fixed to 0
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Second Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	High voltage error generated
	OFF	0x00	High voltage error not generated
1	ON	0x02	Low voltage error generated
	OFF	0x00	Low voltage error not generated
2	ON	0x04	CIS error generated
	OFF	0x00	CIS error not generated
3	-	0x00	Fixed to 0
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Hopper position error generated
	OFF	0x00	Hopper position error not generated
1	ON	0x02	Stamp position error generated
	OFF	0x00	Stamp position error not generated
2	ON	0x04	Switching plate position error generated
	OFF	0x00	Switching plate position error not generated
3	-	0x00	Fixed to 0
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

#### Fourth Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Cut sheet paper stuck error generated
	OFF	0x00	Cut sheet paper stuck error not generated
1	ON	0x02	Cut sheet paper feed error generated
	OFF	0x00	Cut sheet paper feed error not generated
2	ON	0x04	Cut sheet paper length error generated
	OFF	0x00	Cut sheet paper length error not generated
3	ON	0x08	Cut sheet paper transfer error generated
	OFF	0x00	Cut sheet paper transfer error not generated
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

#### Fifth Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Cut sheet paper double feed error generated
	OFF	0x00	Cut sheet paper double feed error not generated
1	ON	0x02	Cut sheet paper insertion direction error generated
	OFF	0x00	Cut sheet paper insertion direction error not generated
2	ON	0x04	External noise error generated
	OFF	0x00	External noise error not generated
3	ON	0x08	Error in the read processing that was designated by a command
	OFF	0x00	No error in the read processing that was designated by a command
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 1
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

**OfflineCode**

## First Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	-	0x00	Fixed to 0
1	-	0x00	Fixed to 0
2	-	0x00	Fixed to 0
3	-	0x00	Fixed to 0
4	OFF	0x00	Power retrigger error not generated (Fixed to 0)
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Second Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	High voltage error generated
	OFF	0x00	High voltage error not generated
1	ON	0x02	Low voltage error generated
	OFF	0x00	Low voltage error not generated
2	-	-	Undefined
3	ON	0x08	FPGA CONFIG error generated
	OFF	0x00	FPGA CONFIG error not generated
4	ON	0x10	Ink-jet mechanical error generated
	OFF	0x00	Ink-jet mechanical error not generated
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Third Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Thermal head thermistor error generated
	OFF	0x00	Thermal head thermistor error not generated
1	ON	0x02	Ink-jet head thermistor error generated
	OFF	0x00	Ink-jet head thermistor error not generated
2	ON	0x04	Waste ink absorption volume error or ink information writing error generated
	OFF	0x00	Waste ink absorption volume error or ink information writing error not generated
3	ON	0x08	Motor driver thermistor error generated
	OFF	0x00	Motor driver thermistor error not generated
4	ON	0x10	Ink-jet head driver thermistor error generated
	OFF	0x00	Ink-jet head driver thermistor error not generated
5	ON	0x20	Pump drive frequency error generated
	OFF	0x00	Pump drive frequency error not generated
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Fourth Byte (Unrecovery Error)

Bit	ON/ OFF	Value	Status
0	-	-	Undefined
1	-	-	Undefined
2	-	-	Undefined
3	-	-	Undefined
4	-	-	Undefined
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Fifth Byte (Recoverable Error: Mechanism position error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Hopper position error generated
	OFF	0x00	Hopper position error not generated
1	ON	0x02	Switching plate position error generated
	OFF	0x00	Switching plate position error not generated
2	-	-	Undefined
3	-	-	Undefined
4	-	-	Undefined
5	ON	0x20	Endorsement printing data not received error generated
	OFF	0x00	Endorsement printing data not received error not generated
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Sixth Byte (Recoverable Error : Roll paper)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Auto-cutter error generated
	OFF	0x00	Auto-cutter error not generated
1	ON	0x02	Roll paper cover open error generated (when specifying recovery error)
	OFF	0x00	Roll paper cover open error not generated (when specifying recovery error)
2	-	-	Undefined
3	-	-	Undefined
4	-	-	Undefined
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Seventh Byte (Recoverable Error: Cut sheet paper)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Cut sheet paper stuck error generated
	OFF	0x00	Cut sheet paper stuck error not generated
1	ON	0x02	Cut sheet paper feed error generated
	OFF	0x00	Cut sheet paper feed error not generated
2	ON	0x04	Cut sheet paper length error generated
	OFF	0x00	Cut sheet paper length error not generated
3	ON	0x08	Cut sheet paper transfer error generated
	OFF	0x00	Cut sheet paper transfer error not generated
4	ON	0x10	Paper path cover open error generated, when cut sheet paper is printed
	OFF	0x00	Paper path cover open error not generated, when cut sheet paper is printed
5	-	-	Undefined
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Eighth Byte (Recoverable Error: Other)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Cut sheet paper double feed error generated
	OFF	0x00	Cut sheet paper double feed error not generated
1	ON	0x02	Cut sheet paper insertion direction error generated
	OFF	0x00	Cut sheet paper insertion direction error not generated
2	ON	0x04	External noise error generated
	OFF	0x00	External noise error not generated
3	ON	0x08	Error in the read processing that was designated by a command
	OFF	0x00	No error in the read processing that was designated by a command
4	ON	0x10	Print specification error caused by specifying the printing content exceeds printable area
	OFF	0x00	Print specification error not caused by specifying the printing content exceeds printable area.
5	ON	0x20	No magnetically read characters error generated
	OFF	0x00	No magnetically read characters error not generated
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

### Ninth Byte (Auto-Recovery Error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Roll paper cover open error generated (when specifying auto-recovery error)
	OFF	0x00	Roll paper cover open error not generated (when specifying auto-recovery error)
1	ON	0x02	Thermal head/ink-jet head driver temporary high-temperature error generated
	OFF	0x00	Thermal head/ink-jet head driver temporary high-temperature error not generated
2	ON	0x04	Motor driver IC temporary high-temperature error generated
	OFF	0x00	Motor driver IC temporary high-temperature error not generated
3	ON	0x08	Ink carriage cover is open
	OFF	0x00	Ink carriage cover is closed
4	ON	0x10	Paper path cover is open
	OFF	0x00	Paper path cover is closed
5	ON	0x20	Roll paper cover is open
	OFF	0x00	Roll paper cover is closed
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Tenth Byte (Auto-Recovery Error)

Bit	ON/ OFF	Value	Status
0	ON	0x01	Thermal head temporary low-voltage error generated
	OFF	0x00	Thermal head temporary low-voltage error not generated
1	-	0x00	Fixed to 0
2	-	0x00	Fixed to 0
3	-	0x00	Fixed to 0
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Device ID

Device ID	Type of Device ID	Status
1	Product ID	<ul style="list-style-type: none"> <li>• 131(83H) for TM-S9000.</li> <li>• 132(84H) for TM-S2000.</li> </ul>
2	Type ID (A)	1-byte data. See <a href="#">Type ID (A) (p.292)</a> .
33	Type information	2-byte data. See <a href="#">Type Information (p.292)</a> .
65	Version of firmware	String data. The obtained data varies depending on the firmware version. Example: "1.00 ESC/POS"
66	Manufacture name	String data. Default is "EPSON". Example: "EPSON"
67	Product name	String data. The obtained data varies depending on the product. Example: "TM-S9000MJ"
68	Serial number	String data. The obtained data varies depending on the device. Example: "DEKK000015"
112	Type ID (B)	1-byte data. See <a href="#">Type ID (B) (p.293)</a> .

### Type ID (A)

Bit	ON/OFF	Value	Status
0	ON	0x01	Multi-byte characters are supported.
	OFF	0x00	Multi-byte characters are not supported.
1	ON	0x02	Auto-cutter is installed. (Fixed to 1 for TM-S9000 and 0 for TM-S2000)
2	OFF	0x00	DM-D is not connected. (Fixed to 0)
3	ON	0x08	MICR reader is installed. (Fixed to 1)
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	ON	0x40	Endorsement printer(E/P) is installed. (Fixed to 1)
7	-	0x00	Fixed to 0

### Type Information

First Byte

Bit	ON/OFF	Value	Status
0	ON	0x01	Multi-byte characters are supported.
	OFF	0x00	Multi-byte characters are not supported.
1	ON	0x02	Auto-cutter is installed. (Fixed to 1 for TM-S9000 and 0 for TM-S2000)
2	OFF	0x00	DM-D is not connected. (Fixed to 0)
3	ON	0x08	MICR reader is installed. (Fixed to 1)
4	ON	0x10	Image scanner is installed. (Fixed to 1)
5	ON	0x20	Endorsement printer(E/P) is installed. (Fixed to 1)
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Second Byte

Bit	ON/ OFF	Value	Status
0	OFF	0x00	No NV memory for saving image scanning results.
1	ON	0x02	Grayscale reading is supported. (Fixed to 1)
2	ON	0x04	Image scanner unit is installed. (Fixed to 1)
3	ON	0x08	Automatic sheet feeder is installed. (Fixed to 1)
4	-	0x00	Fixed to 0
5	ON	0x20	Validation mechanism is installed. (Fixed to 1)
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

**Type ID (B)**

Bit	ON/ OFF	Value	Status
0	ON	0x01	Magnetic stripe reader is installed.
	OFF	0x00	Magnetic stripe reader is not installed.
1	ON	0x02	2 pockets
	OFF	0x00	1 pocket
2	-	0x04	Fixed to 1
3	Maximum speed setting status (See <a href="#">Maximum speed setting status (p.293).</a> )		
4			
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

## Maximum speed setting status

	3 Bit	4 Bit
110DPM	ON (0x01)	OFF (0x00)
200DPM	OFF (0x00)	ON (0x01)

## Type ID

Bit	ON/ OFF	Value	Status
0	ON	0x01	Multi-byte characters are supported.
	OFF	0x00	Multi-byte characters are not supported.
1	ON	0x02	Auto-cutter is installed. (Fixed to 1 for TM-S9000 and 0 for TM-S2000)
2	OFF	0x00	DM-D is not connected. (Fixed to 0)
3	ON	0x08	MICR reader is installed. (Fixed to 1)
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	ON	0x40	Endorsement printer(E/P) is installed. (Fixed to 1)
7	-	0x00	Fixed to 0