EPSON
EXCEED YOUR VISION

Advanced Printer Driver Ver.4

Status API Manual

Overview

Explains about Status API.

Using Status API

Explains how to establish a development environment, get ASB statuses, and how to handle the ASB statuses.

Using the MICR/Scanner

Explains how to use functions of the MICR/scanner using Status API.

Reference for Win32

Describes available Status API and the syntax for each printer model.

Reference for .NET

Describes information of Status API used in .NET environment.

Generating Log Files

Describes the log function.

Appendix

Describes information of each TM printer acquired by Status API.

M00002505
Rev.G

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.

Trademarks

EPSON® and ESC/POS® are registered trademarks of Seiko Epson Corporation in the U.S. and other countries.

MS-DOS®, Microsoft®, Win32®, Windows®, Windows Vista®, Windows 7™, Windows Server®, Visual Studio®, Visual Basic®, Visual C++®, and Visual C#® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

ESC/POS® Command System

EPSON has been taking industry's initiatives with its own POS printer command system (ESC/POS). ESC/POS has a large number of commands including patented ones. Its high scalability enables users to build versatile POS systems. The system is compatible with all types of EPSON POS printers (excluding the TM-C100) and displays. Moreover, its flexibility makes it easy to upgrade the future. The functionality and the user-friendliness is valued around the world.

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.

CAUTION	Provides information that must be observed to avoid damage to your equipment or a malfunction.
NOTE	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 engineer and provides necessary information for developing an application using the Status API.

Manual Content

The manual is made up of the following sections:

Chapter 1	Overview
Chapter 2	Using Status API
Chapter 3	Using the MICR/Scanner
Chapter 4	Reference for Win32
Chapter 5	Reference for .NET
Chapter 6	Generating Log Files
Appendix	Model Information

Contents

■ **For Safety 3**

Key to Symbols 3

■ **Restriction of Use 3**

■ **About this Manual 4**

Aim of the Manual 4
Manual Content 4

■ **Contents 5**

Overview 13

Manual organization 13

■ **Status API Summary 14**

Status API System 14
Glossary 14

■ **Information that can be Acquired from the TM Printer 15**

■ **Development Language 15**

Using Status API 17

■ **Install and Uninstall 17**

■ **Architecture of the Development Environment 17**

■ **Types of Status API Functions 19**

■ **Acquiring ASB Status 21**

BiGetStatus 22
BiSetStatusBackFunction 23
BiSetStatusBackWnd 24

■ **Status API Errors and Response 25**

ASB Status 25
Status API Execution Error 27

■ **How to Use Shared Printers 29**

Constructing the Exclusive Access 29
When Using APD3.xx Application 30

Using the MICR/Scanner 31

■ Using the MICR 32
■ Using the Scanner 34
■ Image data process 36

Reference for Win32...41

■ Status API used for each model 41
■ BiOpenMonPrinter 59
■ BiCloseMonPrinter 61
■ BiLockPrinter 62
■ BiUnlockPrinter.. 64
■ BiSetMonInterval 65
■ BiSetMonEtherInterval 66
■ BiDirectIO 67
■ BiDirectIOEx 69
■ BiResetPrinter 73
■ BiCancelError 75
■ BiGetType... 77
■ BiGetOfflineCode 78
■ BiGetStatus... 79
■ BiGetRealStatus 80
■ BiSetStatusBackFunction... 82
■ BiSetStatusBackFunctionEx... 84
■ BiSetStatusBackWnd 86
■ BiCancelStatusBack... 87
■ BiGetInkStatus.. 88
■ BiSetInkStatusBackFunction 89
■ BiSetInkStatusBackFunctionEx 91
■ BiSetInkStatusBackWnd 93
■ BiCancelInkStatusBack 94
■ BiGetBatteryStatus... 95
■ BiSetBatteryStatusBackFunctionEx 97
■ BiSetBatteryStatusBackWndEx 99
■ BiCancelBatteryStatusBack 100
■ BiSetPresenterEventBackFunction 101

■ BiSetPresenterEventBackFunctionEx 103
■ BiSetPresenterEventBackWnd 104
■ BiCancelPresenterEventBack 105
■ BiAutoPowerOffTime 106
■ BiPowerOff 108
■ BiEjectSheet 110
■ BiGetCounter 111
■ BiResetCounter 113
■ BiGetPrnCapability 115
■ BiOpenDrawer 117
■ BiPreparePaperChange 119
■ BiRetractSheet 120
■ BiSendDataFile 121
■ BiDirectSendRead 123
■ BiSetDefaultEchoTime 126
■ BiSetEtherEchoTime 127
■ BiSetReadWaitTimeOut 128
■ BiMICRCleaning 129
■ BiMICRGetStatus 130
■ BiMICRLoadCheck 132
■ BiMICRReadCheck 133
■ BiMICREjectCheck 135
■ BiMICRCancelWaitCheckInsertion 136
■ BiMICRRetransmissionCheckData 137
■ BiMICRSelectDataHandling 139
■ BiMICRSetReadBackFunction 141
■ BiMICRSetReadBackFunctionEx 144
■ BiMICRSetReadBackWnd 146
■ BiMICRCancelReadBack 148
■ BiSCNClumpPaper 149
■ BiSCNGetClumpStatus 150
■ BiSCNGetCroppingArea 152
■ BiSCNSetCroppingArea 154
■ BiSCNDeleteCroppingArea 156
■ BiSCNGetImageFormat 157
■ BiSCNSetImageFormat 159
■ BiSCNGetImageList 161

■ BiSCNGetImageQuality 163
■ BiSCNSetImageQuality 165
■ BiSCNGetImageRemainingCapacity 167
■ BiSCNGetScanArea 169
■ BiSCNSetScanArea 171
■ BiSCNPreScan 174
■ BiSCNReadImage 176
■ BiSCNRetransmissionImage 179
■ BiSCNDeleteImage 181
■ BiSCNSelectScanUnit 183
■ BiSCNSetReadBackFunction 185
■ BiSCNSetReadBackFunctionEx 188
■ BiSCNSetReadBackWnd 190
■ BiSCNCancelReadBack 192
■ BiESCNEnable 193
■ BiESCNGetAutoSize 194
■ BiESCNSetAutoSize 195
■ BiESCNGetCutSize 196
■ BiESCNSetCutSize 197
■ BiESCNGetRotate 198
■ BiESCNSetRotate 199
■ BiESCNGetDocumentSize 200
■ BiESCNSetDocumentSize 201
■ BiESCNDefineCropArea 202
■ BiESCNGetMaxCropAreas 204
■ BiESCNStoreImage 205
■ BiESCNRetrieveImage 208
■ BiESCNClearImage 211
■ BiESCNGetRemainingImages 213

Reference for .NET..215

■ Properties..215

AutoPowerOffTime 215
ESCNAutoSize 215
ESCNCutSize 215
ESCNDocumentSize 216
ESCNRemainingImages 216
ESCNRotate 216
InkStatus 217
IsValid 217
LastError 217
OfflineCode 218
Status 218

■ Methods..219

OpenMonPrinter 219
CloseMonPrinter 219
LockPrinter 220
UnlockPrinter 220
SetMonInterval 220
SetMonEtherInterval 221
DirectIOEx 221
ResetPrinter 222
CancelError 222
GetType 222
GetRealStatus 223
SetStatusBack 223
CancelStatusBack 223
SetInkStatusBack 224
CancelInkStatusBack 224
GetBatteryStatus 224
SetBatteryStatusBack 225
CancelBatteryStatusBack 225
SetPresenterEvent 225
CancelPresenterEvent 226
PowerOff 226
EjectSheet 226
GetCounter 227
ResetCounter 227
GetPrnCapability 228
OpenDrawer 228
PreparePaperChange 229
RetractSheet 229
SendDataFile 230
DirectSendRead 231
SetDefaultEchoTime 232
SetEtherEchoTime 232
MICRCleaning 233
MICRGetStatus 233
MICRLoadCheck 234

MICRReadCheck 234
MICREjectCheck 234
MICRCancelWaitCheckInsertion 235
MICRRetransmissionCheckData 235
MICRSelectDataHandling 236
MICRSetReadBack 236
MICRCancelReadBack 237
SCNClumpPaper 237
SCNGetClumpStatus 237
SCNGetCroppingAreas 238
SCNSetCroppingArea 238
SCNDeleteCroppingArea 239
SCNGetImageFormat 239
SCNSetImageFormat 240
SCNGetImageList 240
SCNGetImageQuality 241
SCNSetImageQuality 242
SCNGetImageRemainingCapacity 242
SCNGetScanArea 243
SCNSetScanArea 243
SCNPreScan 244
SCNReadImage 244
SCNRetransmissionImage 245
SCNDeleteImage 245
SCNSelectScanUnit 246
SCNSetReadBack 246
SCNCancelReadBack 246
ESCNEnable 247
ESCNDefineCropArea 247
ESCNGetMaxCropAreas 248
ESCNStoreImage 248
ESCNRetrieveImage 249
ESCNClearImage 249

■ **Events 250**

StatusCallback 250
StatusCallbackEx 250
InkStatusCallback 251
InkStatusCallbackEx 251
BatteryStatusCallbackEx 252
MICRReadCallback 252
MICRReadCallbackEx 253
SCNReadCallback 253
SCNReadCallbackEx 254
PresenterCallback 255
PresenterCallbackEx 255

Generating Log Files 257

■ **Log Files Settings 258**

■ **Viewing Log Files 259**

Appendix..261

■ Model Information..261

BA-T500.. 261
EU-T300.. 263
EU-T400.. 264
EU-T500.. 267
TM-C100.. 270
TM-H5000II.. 276
TM-H5200.. 278
TM-H6000II.. 280
TM-H6000III.. 284
TM-J2100.. 288
TM-J7100.. 293
TM-J7500/7600.. 298
TM-L60II.. 303
TM-L90.. 304
TM-L90Peeler.. 308
TM-P60.. 312
TM-P60 Peeler.. 314
TM-T70.. 316
TM-T88III.. 318
TM-T88IV.. 319
TM-T88IV ReStick.. 321
TM-T90.. 323
TM-U120.. 327
TM-U120II.. 329
TM-U220.. 331
TM-U230.. 332
TM-U295.. 333
TM-U375.. 334
TM-U590.. 336
TM-U675.. 337
TM-U950.. 339

Overview

Status API is a status monitor API for Epson's TM printers. Advanced functions for monitoring TM printers can be embedded in applications with print functions.

Manual organization

Install Manual

Descriptions of the procedures from installing the APD to performing test print, adding printer drivers, and the silent install which is an automated APD installation.

TM Printer Manual

Descriptions of how to use the APD and its functions.

Printer Specification

Descriptions of the specifications of each printer driver.

TM-C100 Manual

Descriptions of how to use the TM-C100 printer driver and its functions.

Customer Display Manual

Descriptions of how to use the Customer Display printer driver and its functions.

Status API Manual

This manual. Descriptions of how to get the status of the TM printer from the user application by using the Status API.

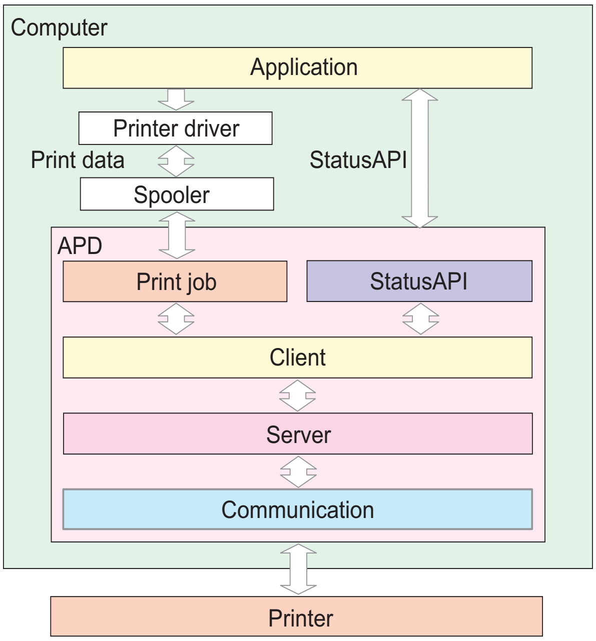
Devmode API / PRINTERINFO Manual

Descriptions of how to configure some printer functions on your application using the Devmode API. Descriptions of the PRINTERINFO Function of Windows.

Status API Summary

Status API System

Status API receives signals from the TM printer and always maintains the most recent status of the printer. The application can acquire the most recent information when necessary.



CAUTION

- In Terminal Service / Citrix XenApp environment, Status API cannot be used.
- Status API cannot be used when using the TM-C100 printer via the standard TCP/IP port.

Glossary

Term	Explanation
APD	Advanced Printer Driver: Windows printer driver for TM printers. Unlike general Windows printer drivers, Status API is simultaneously installed to monitor the printer status.
ASB Status	Auto Status Back: This is a function of the TM printer. This is a status automatically sent from the printer when the printer status changes (opening or closing the cover, out of paper, print completed, etc.).
Maintenance Counter	TM printer internal counter recording the operating status of the printer, i.e. operating count of the auto cutter, printer running time, etc.

Information that can be Acquired from the TM Printer

Term	Explanation
ASB Status	Information required for the print application, i.e. print completed, offline, out of paper, cover open, power off, error generated, etc. This information is automatically sent to the Status API.
Ink ASB Status	Ink related information, such as Ink level (not an indicator of ink level, but is a notification that the ink cartridge needs to be replaced because the ink level is low.), ink cartridge existence, that is necessary for a print application for an inkjet printer. In the same way as the ASB Status, this information is automatically sent to the Status API from the TM printer.
Maintenance Counter	Acquires information, i.e number of feed paper lines, operating count of the auto cutter, running time, etc. This is used for printer management applications. There are counters that can be reset from Status API and there are integral counters that cannot be reset.

CAUTION

According to the printer model, these functions may not be installed.
Refer to "Model Information" on page 261 for details.

Development Language

Win32

- Visual Basic 6.0
- Visual C++

.NET

- Visual Basic .NET
- Visual C#

CAUTION

.NET Framework Version

Conforming to the APD environment. Refer to "Install Manual".
If you use Status API .NET Wrapper in Windows XP, install .NET Frame Work 2.0 or later before installing APD.

Using Status API

This chapter explains the architecture of the application development environment using Status API, the acquisition methods of ASB Status, and the procedures of the ASB Status. Refer to "Reference for Win32" on page 41 for other functions.

Install and Uninstall

Status API is installed/uninstalled at the same time as APD (Advanced Printer Driver) is installed/uninstalled. Refer to the "Install Manual" for details.

Architecture of the Development Environment

The architecture of the application development environment using Status API differs according to the development tool.

Visual Basic

The following are examples of the development environment architecture using Visual Basic.

1 Copy StatusAPI.bas in the folder where the sample program is installed (default is C:\Program Files\EPSON\EPSON Advanced Printer Driver 4 \Sample\US\Src\VB6\SingleFunction\Program09") and paste it into the operating folder used when developing applications.

2 Start Microsoft Visual Basic and open the project screen.

3 Select (Add a standard module) from (Project) in the menu bar.

4 The add standard module screen appears. Select the (Existing file) tab and specify "StatusAPI.bas" copied from the sample program in Procedure 1. Click the (Open) button.
"StatusAPI.bas" is added the project explorer.

5 Select (Reference settings) from (Project) in the menu bar.

6 The reference settings screen appears. Place a check by "Microsoft DAO 3.6 Object Library" from the (Reference library file) and click the (OK).

C++

The following are examples of the development environment architecture using C++.

1 Start Microsoft Visual C++ and open the project screen.

2 Copy EpsStmApi.h from the folder installed with APD and paste the file into the operating folder used when developing applications (folder created by the project).

3 Open the Source File. Define EpsStmApi.h using the #include directive.
Definition Methods: *#include "EpsStmApi.h"*

Types of Status API Functions

Status API has the following functions. Refer to "Reference for Win32" on page 41 for details regarding the functions. The supported functions differ according to the printer model. Refer to "Reference for Win32" on page 41 for details regarding each model.

Application	Function	Description
Starting/Closing Status API	BiOpenMonPrinter	Calls the specified printer to use Status API.
	BiCloseMonPrinter	Closes Status API.
Acquiring ASB Status	BiGetStatus	Acquires the ASB status from Status API when required by the application.
	BiSetStatusBackFunction	Provides notification regarding the call of the callback function notifying the application when the ASB status of Status API changes.
	BiSetStatusBackFunctionEx	Provides notification regarding the call of the callback function notifying the application when the ASB status of Status API changes. Also acquires the port number.
	BiSetStatusBackWnd	Generates a button click event when the ASB status of Status API changes.
	BiCancelStatusBack	Cancels the auto status notification function. This function is applicable to BiSetStatusBackFunction, BiSetStatusBackFunctionEx, and BiSetStatusBackWnd.
Acquiring and resetting the maintenance counter	BiGetCounter	Acquires the maintenance counter value of the printer.
	BiResetCounter	Resets the maintenance counter of the printer.
Acquiring the printer information	BiGetPrnCapability	Acquires printer information, i.e. firmware, etc.
Drawer control	BiOpenDrawer	Opens the drawer.
Recovery from a recoverable error	BiCancelError	After the cause of the error, i.e. paper jam, etc., is removed, the printer's auto cutter is recovered from the error status using this function. The status is recovered to print standby without turning the printer's power off and on.
Printer reset	BiResetPrinter	Resets the parallel / USB / ethernet interface printers. Cannot reset serial interface printers.
Power off preprocess	BiPowerOff	Updates the maintenance counter.

Application	Function	Description
Command definition file	BiSendDataFile	Defines the command definition file created separately to the printer. The command is not executed.
	BiDirectSendRead	Sends the command of the defined command definition file to the printer and executes the command.
Sends the ESC/POS command	BiDirectIO	Can transmit ESC/POS commands to the printer and receive data from the printer.
	BiDirectIOEx	Can send and receive the ESC/POS commands. Does not add the ASB suppress command.

Acquiring ASB Status

The method and function to acquire ASB status from the application are as follows.

Timing	Status API
Acquires when required by the application	BiGetStatus
The ASB status is acquired as follows	BiSetStatusBackFunction
	BiSetStatusBackFunctionEx
	BiSetStatusBackWnd

The ASB status is acquired as follows.

- Confirms whether the printer can print in advance.
- Confirms that printing has completed successfully. Confirms with “ASB_PRINT_SUCCESS” (constant) of the macro definition.
- ASB status monitors the main printer conditions, i.e. out-of-paper, cover open, printer connection status, etc. Therefore, it is recommended that the printer is continually monitored, even when not printing.

Refer to "ASB Status" on page 25 regarding acquired ASB status.

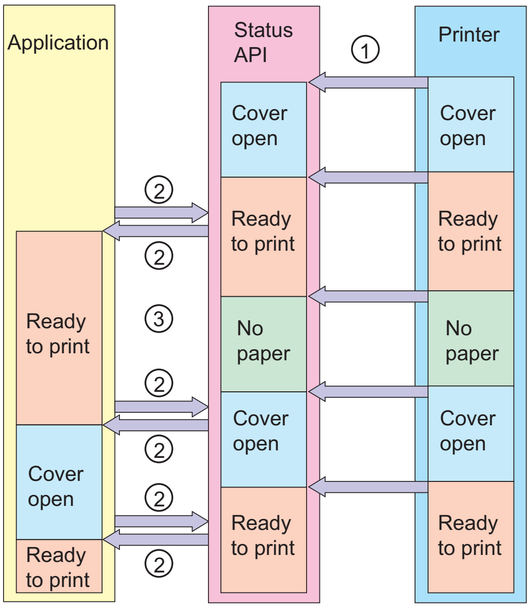
NOTE

Status API has to be started using BiOpenMonPrinter and the printer needs to be opened when acquiring ASB status.

BiGetStatus

BiGetStatus acquires ASB status when required by the user (or application).

Example: The following diagram explains the flow of Status API and ASB status using BiGetStatus.



[1] The printer automatically sends the ASB status to Status API using the ASB function each time the status changes. Status API stores the most recent ASB status.

[2] The application calls BiGetStatus when required by ASB status. Status API sends the stored ASB status to the application.

[3] Status API does not send the ASB status even if the ASB status of the printer changes when there is no request from the application.

Refer to "BiGetStatus" on page 79 regarding the syntax of BiGetStatus.

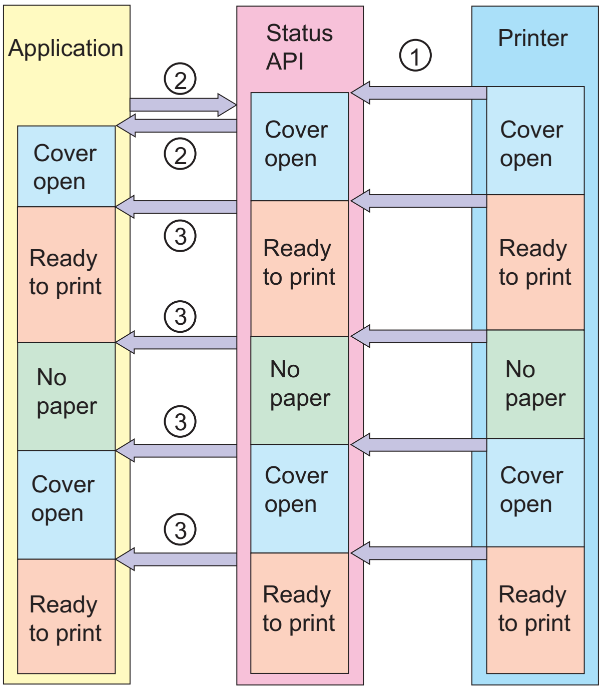
BiSetStatusBackFunction

BiSetStatusBackFunction is an API that automatically allows the application to acquire the most recent ASB status by using the callback function.
Calling BiCancelStatusBack cancels the ASB status notification from Status API using the callback function. Refer to "BiCancelStatusBack" on page 87 for details.

CAUTION
This is unavailable when the development environment is Visual Basic.

NOTE
BiSetStatusBackFunctionEx can recognize from which printer the callback is, in addition to the function of BiSetStatusBackFunction.

Example: The following diagram explains the flow of Status API and ASB status using BiSetStatusBackFunction.



[1] The printer automatically sends the ASB status to Status API using the ASB function each time the status changes.

[2] BiSetStatusBuckFunction registers the callback function.

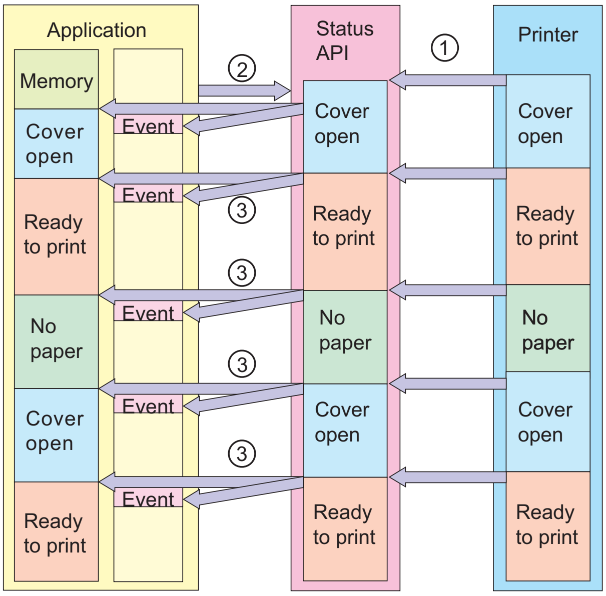
[3] Status API calls the callback function and notifies the application of the most recent ASB status each time the ASB status of the printer changes.

Refer to "BiSetStatusBackFunction" on page 82 regarding the syntax of BiSetStatusBackFunction.

BiSetStatusBackWnd

BiSetStatusBackWnd is an API that acquires the most recent ASB status by registering the window handle of the application screen button and memory address storing the status. Calling BiCancelStatusBack cancels the ASB status notification from Status API using the button click event. Refer to "BiCancelStatusBack" on page 87 for details.

Example: The following diagram explains the flow of Status API and ASB status using BiSetStatusBackWnd.



[1] The printer automatically sends the ASB status to Status API using the ASB function each time the status changes. Status API stores the most recent ASB status.

[2] When BiSetStatusBackWnd registers the window handle of the application screen button and memory address storing the status, Status API sets the data to the specified address and sends a button click event.

[3] Status API sets the most recent ASB status of the printer to the specified memory and sends a button click event each time the ASB status of the printer changes.

Refer to "BiSetStatusBackWnd" on page 86 regarding the syntax of BiSetStatusBackWnd.

Status API Errors and Response

Status API errors are errors when notifying ASB status and errors generated when calling Status API. The following explains the details of the errors and responses. Refer to the following and respond to the application errors.

ASB Status

The following are errors returned when ASB status is acquired. The details differ according to the printer model. Refer to "Model Information" on page 261 for details.

Macro Definition (Constant)	Cause	Response
ASB_NO_RESPONSE	The power to the printer is not turned ON. The communication cable is not connected. The specified printer name/port is different.	Confirm the status and ports of the printer, i.e. cables, etc.
ASB_PRINT_SUCCESS	Notifies that printing has completed successfully. There is nothing else that is notified.	-
ASB_DRAWER_KICK	The drawer is open.	There is no problem if the drawer has been left open intentionally.
ASB_OFF_LINE	An error causing the printer to go offline was generated.	Eliminate the cause of the printer to go offline.
ASB_COVER_OPEN	The cover is open.	Close the printer's cover.
ASB_PAPER_FEED	Paper is being fed.	There is no problem if the paper is being fed.
ASB_WAIT_ON_LINE	Waiting for getting back into online.	-
ASB_PANEL_SWITCH	The printer's panel switch is being pressed.	There is no problem if a switch on the printer's panel is being pressed.
ASB_MECHANICAL_ERR	A mechanical error was generated, i.e. home position detection error, etc.	Eliminate the cause of the error and restart the printer or send an error recovery command (BiCancelError).*
ASB_AUTOCUTTER_ERR	An auto cutter error was generated.	Eliminate the cause of the error and restart the printer or send an error recovery command (BiCancelError).*

Macro Definition (Constant)	Cause	Response
ASB_UNRECOVER_ERR	A print error was generated to the printer.	Immediately turn off the power to the printer.*
ASB_AUTORECOVER_ERR	The temperature of the head has increased.	If the temperature of the head decreases with time, the error automatically cancels.*
ASB_RECEIPT_NEAR_END	There is only a limited amount of paper remaining.	Place paper in the printer.
ASB_RECEIPT_END	No paper.	Place paper in the printer.
ASB_SLIP_TOF	No paper is detected. This is not an error but indicating the printer status.	-
ASB_SLIP_BOF	No paper is detected. This is not an error but indicating the printer status.	-
ASB_SLIP_SELECTED	This indicates whether the slip has been selected.	-
ASB_PRINT_SLIP	The printer is waiting for the slip to be inserted or the slip is being copied/discharged.	-
ASB_VALIDATION_SELECTED	This indicates whether the validation has been selected.	-
ASB_PRINT_VALIDATION	The printer is waiting for the slip to be inserted or the validation is being copied/discharged.	-
ASB_VALIDATION_TOF	No paper is detected. This is not an error but indicating the printer status.	-
ASB_VALIDATION_NO_PAPER	No paper is detected by the validation detector.	-

* Refer to the detailed operating manuals of the various printers.

NOTE

The macro definition is defined using the EPSStmApi.h or StatusAPI.bas file when the development environment is constructed.

Status API Execution Error

The following are errors generated when Status API functions are called. The contents differ according to Status API function. Refer to "Model Information" on page 261 for details.

Macro Definition (Constant)	Cause	Response
ERR_TYPE	The parameters of nType differ.	Specify the correct value.
ERR_OPENED	The specified printer is already opened.	As the printer is already opened, use the handle value or specify a different printer.
ERR_NO_PRINTER	The specified printer driver does not exist.	Confirm the name of the printer driver.
ERR_NO_TARGET	The specified printer cannot be found. An unspecified printer is connected.	Connect to the correct printer.
ERR_NO_MEMORY	There is not enough memory.	Add available memory.
ERR_HANDLE	The handle value specified by the printer is incorrect.	Confirm the handle value.
ERR_TIMEOUT	This is a timeout error.	If the error is continuously generated, confirm whether the printer is properly connected.
ERR_ACCESS	R/W cannot be performed on the printer. (The power to the printer is not turned on or the cable is not properly connected, etc.)	Confirm the printer. (Printer power, cable connection, etc.)
ERR_PARAM	This is a parameter error.	Review the syntax as the specified parameter is incorrect.
ERR_NOT_SUPPORT	This is an unsupported model.	Unsupported models cannot be used.
ERR_EXIST	The specified data already exists.	Delete the already existing data. Example: When this error occurs during executing BiSetStatusBackXXX, retry it after executing BiCancelStatusBack.
ERR_EXEC_FUNCTION	This function is unavailable as Status API is used by other applications.	Close the Status API used by other applications.
ERR_PH_NOT_EXIST	The PortHandler is not running, or a communication error between the client of PortHandler and the server.	Verify the connection between them, then restart the computer.

Macro Definition (Constant)	Cause	Response
ERR_SPL_NOT_EXIST	The spooler service is not operating.	Confirm whether the Print Spooler is properly operating. (Control Panel - Management Tools - Service)
ERR_RESET	This function is unavailable as the printer is being reset.	Recall after waiting a moment.
ERR_LOCKED	The printer is locked.	Wait until the printer becomes unlocked, or execute BiUnlockPrinter in the program that is locking the printer.

NOTE

The macro definition is defined using the EPSStmApi.h or StatusAPI.bas file when the development environment is constructed.

How to Use Shared Printers

When using shared printers, note the followings when developing an application.

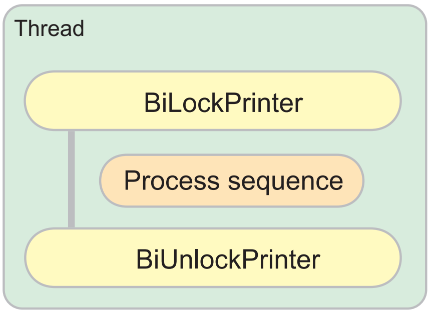
- A whole process of accessing to the device shall be performed exclusively.
- Handling of BiLockPrinter error is necessary.
- Set a time of exclusive access to the printer as short as possible.

CAUTION When you set a firewall, add the port number 2291 to [Exception].

Constructing the Exclusive Access

Exclusive control of a process sequence

Construct an application putting each of process sequences between BiLockPrinter and BiUnlockPrinter.



NOTE When you use only one API in your application, the exclusive access is not necessary.

Handling of BiLockPrinter error

When the printer is already accessed from another process, the BiLockPrinter returns an error (ERR_LOCKED). Construct the application so that the error is handled and the BiLockPrinter is executed again after handling the error.

Shortening exclusive access time to the printer

While the printer is accessed exclusively, other processes cannot execute exclusive API and printing is disabled.
Therefore, set the exclusive access time as short as possible to improve the system performance.

Program example

```
int nRet = BiLockPrinter(1, 1000);
if (nRet == SUCCESS) {
 //Locks the printer to allow exclusive access of the following API. Unlocks the printer when finished.
BiSCNSetImageFormat(1, EPS_BI_SCN_JPEGNORMAL);
BiSCNReadImage(1, 1, EPS_BI_SCN_CHECKPAPER, 0, 0, NULL, EPS_BI_SCN_NVMEMORY_NOTSAVE);
//Unlocks the printer
BiUnlockPrinter(1);
} else {
 //Error handling in case that the exclusive access is failed.

}
```

When Using APD3.xx Application

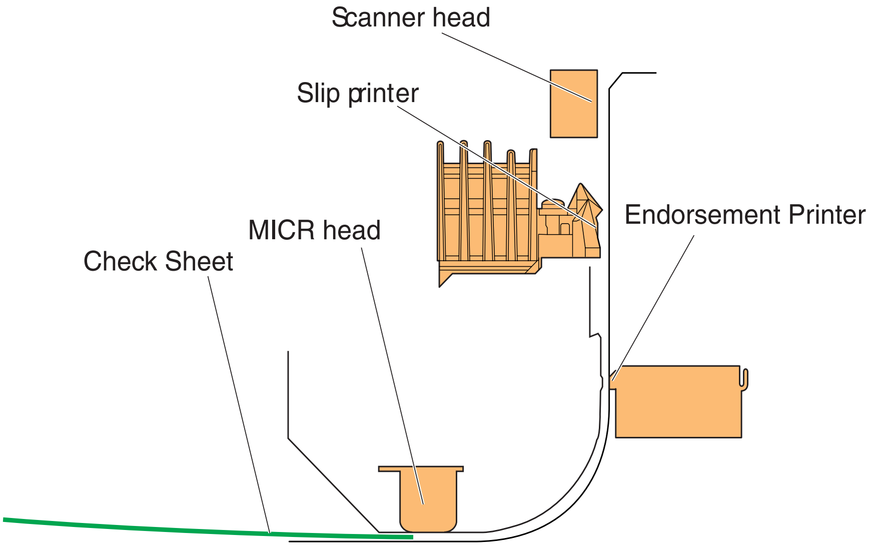
The APD3.xx does not support the exclusive access to the printer and consequently your application that supports the APD3.xx do not have the function for the exclusive access . Therefore, when you use the existing application and do not use any other one, the exclusive access to the printer is available without modifying the application.

If any access from other processes is expected, modify the application to establish the exclusive access referring to "Constructing the Exclusive Access" on page 29.

Using the MICR/Scanner

This chapter describes the functions built into the MICR/Scanner using Status API.

The following printer mechanism achieves the MICR/Scanner function.

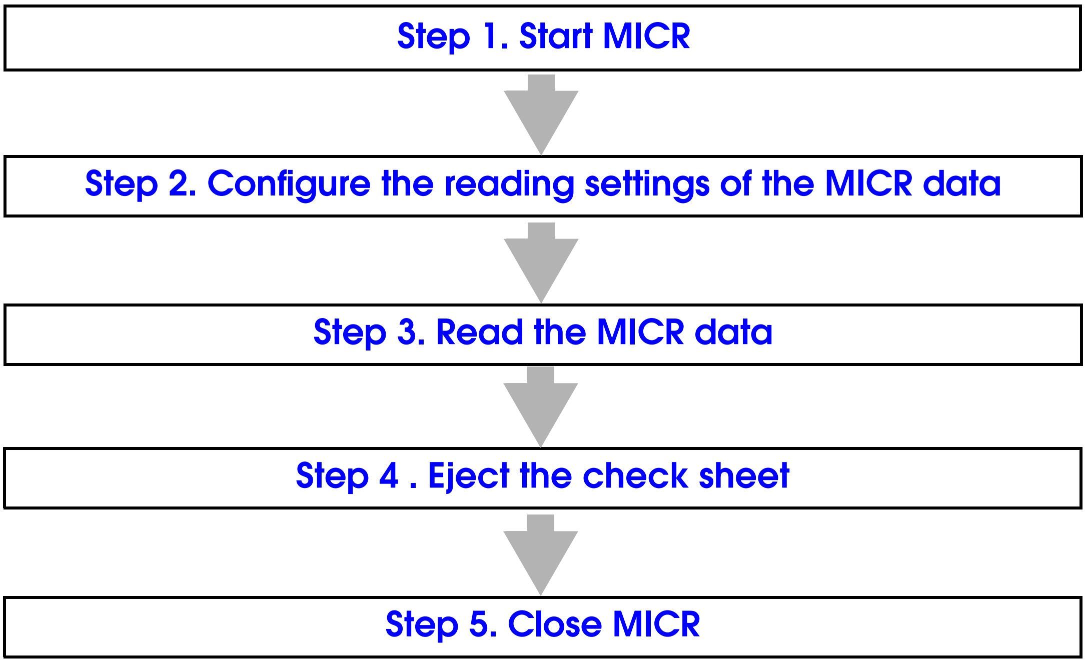


NOTE

A single-sequence sample program that performs the following with one motion using functions of the hybrid printer with Status API is provided: reading MICR data, endorsement printing, front printing, and reading image data.

Using the MICR

The MICR is a process that reads magnetic characters. Call Status API in the following flow to use MICR.



Step 1. Start MICR

Step 2. Configure the reading settings of the MICR data

Step 3. Read the MICR data

Step 4 . Eject the check sheet

Step 5. Close MICR

Step 1. Start MICR

Call the following Status API to acquire MICR data. For details regarding Status API, refer to “Reference for Win32” on page 41.

Status API	Page	Description
BiMICRSetReadBackFunction	141	The MICR read data is sent to an application using a callback function or button-click event.
BiMICRSetReadBackFunctionEx	144	
BiMICRSetReadBackWnd	146	

Step 2. Configure the reading settings of the MICR data

Configure the reading settings when reading MICR data by calling BiMICRSelectDataHandling. BiMICRSelectDataHandling is called as follows. For details regarding Status API, refer to "BiMICRSelectDataHandling" on page 139.

nErr = BiMICRSelectDataHandling (nHandle, charSelect, detailSelect, errorSelect)

Step 3. Read the MICR data

Call BiMICRReadCheck to acquire the MICR data. Configure the read fonts (*readFont*) and check sheet insertion wait time (*waitInsertionTime*) and call as follows. Refer to "BiMICRReadCheck" on page 133.

nErr = BiMICRReadCheck (nHandle, readFont, waitInsertionTime)

When Status API acquires MICRdata, the callback function or button-click event configured in "Step 1. Start MICR" on page 32 is generated. The MICR data is confirmed here for the first time.

NOTE

If BiMICRSetReadBackFuncion, BiMICRSetReadBackFuncionEx, or BiMICRSetReadBackWnd is not called, a Status API execution error is returned. MICR data cannot be acquired.

Step 4 . Eject the check sheet

Call BiMICREjectCheck to eject the read check sheet. BiMICREjectCheck is called as follows. For details regarding Status API, refer to "BiMICREjectCheck" on page 135.

nErr = BiMICREjectCheck (nHandle)

Step 5. Close MICR

Call BiMICRCancelReadBack to close MICR. BiMICRCancelReadBack is called as follows. For details regarding Status API, refer to "BiMICRCancelReadBack" on page 148.

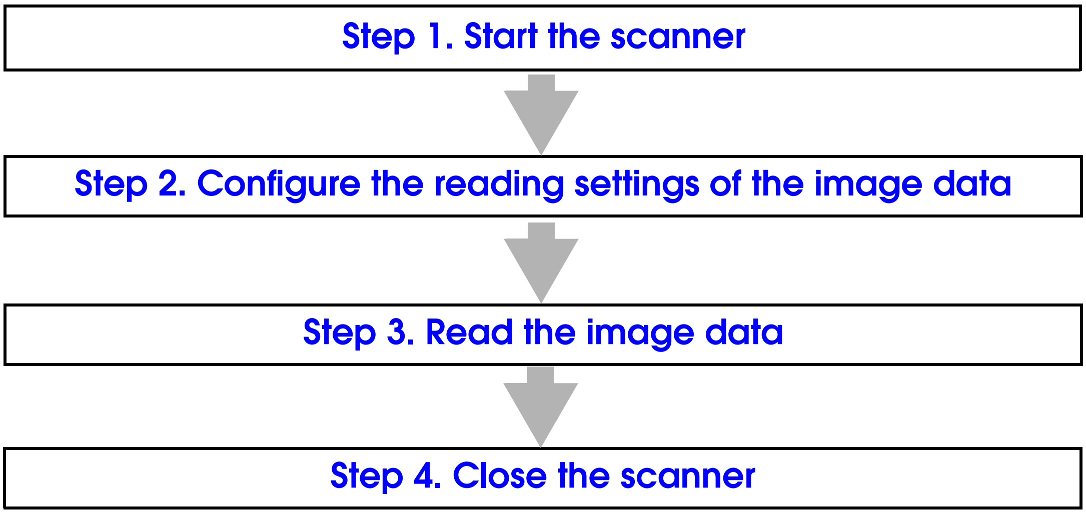
NOTE

If this Status API is called, MICR data cannot be acquired even if BiMICRReadCheck is called. To acquire MICR data after calling Status API, recall BiMICRSetReadBackFuncion, BiMICRSetReadBackFuncionEx, and BiMICRSetReadBackWnd.

nErr = BiMICRCancelReadBack (nHandle)

Using the Scanner

The scanner uses a process to read image data. Call Status API in the following flow to use the scanner.



Step 1. Start the scanner

Step 2. Configure the reading settings of the image data

Step 3. Read the image data

Step 4. Close the scanner

Step 1. Start the scanner

Call the following Status API to acquire the image data. For details regarding Status API, refer to "Reference for Win32" on page 41.

Status API	Page	Description
BiSCNSetReadBackFunction	185	The read image data is sent to an application using a callback function or button-click event.
BiSCNSetReadBackFunctionEx	188	
BiSCNSetReadBackWnd	190	

Step 2. Configure the reading settings of the image data

Call the following Status API to acquire the image data using the scanner. For details regarding Status API, refer to "Reference for Win32" on page 41.

Status API	Page	Description
BiSCNSetImageQuality	165	Configures the reading quality of the image data.
BiSCNSetImageFormat	159	Configures the format of the reading image data.
BiSCNSetScanArea	171	Configures the reading area of the image data.
BiSCNPreScan	174	Automatically measures and configures the density of the image data. This Status API is called when BiSCNSetImageQuality is configured to 2 tone.
BiSCNSetCroppingArea	154	Acquires the specified area of the image data read by BiSCNReadImage.

Step 3. Read the image data

Call BiSCNReadImage to acquire the image data. BiSCNReadImage is called as follows. Refer to "BiSCNReadImage" on page 176.

nErr = BiSCNReadImage (nHandle, wId, bSelectSheet, bWaitInsertionTime, bAddInforDataSize, pAddInforData, bMemory)

When Status API acquires the image data, the callback function or button-click event configured in "Step 1. Start the scanner" on page 34 is generated. The image data is confirmed here for the first time.

NOTE

If BiSCNSetReadBackFuncion, BiSCNSetReadBackFuncionEx, or BiSCNSetReadBackWnd is not called, a Status API execution error is returned. The image data cannot be acquired.

Step 4. Close the scanner

Call BiSCNCancelReadBack to close the scanner. BiSCNCancelReadBack is called as follows. For details regarding Status API, refer to "BiSCNCancelReadBack" on page 192.

NOTE

If this Status API is called, the image data cannot be acquired even if BiSCNReadImage is called. To acquire the image data after calling Status API, recall BiSCNSetReadBackFuncion, BiSCNSetReadBackFuncionEx, or BiSCNSetReadBackWnd.

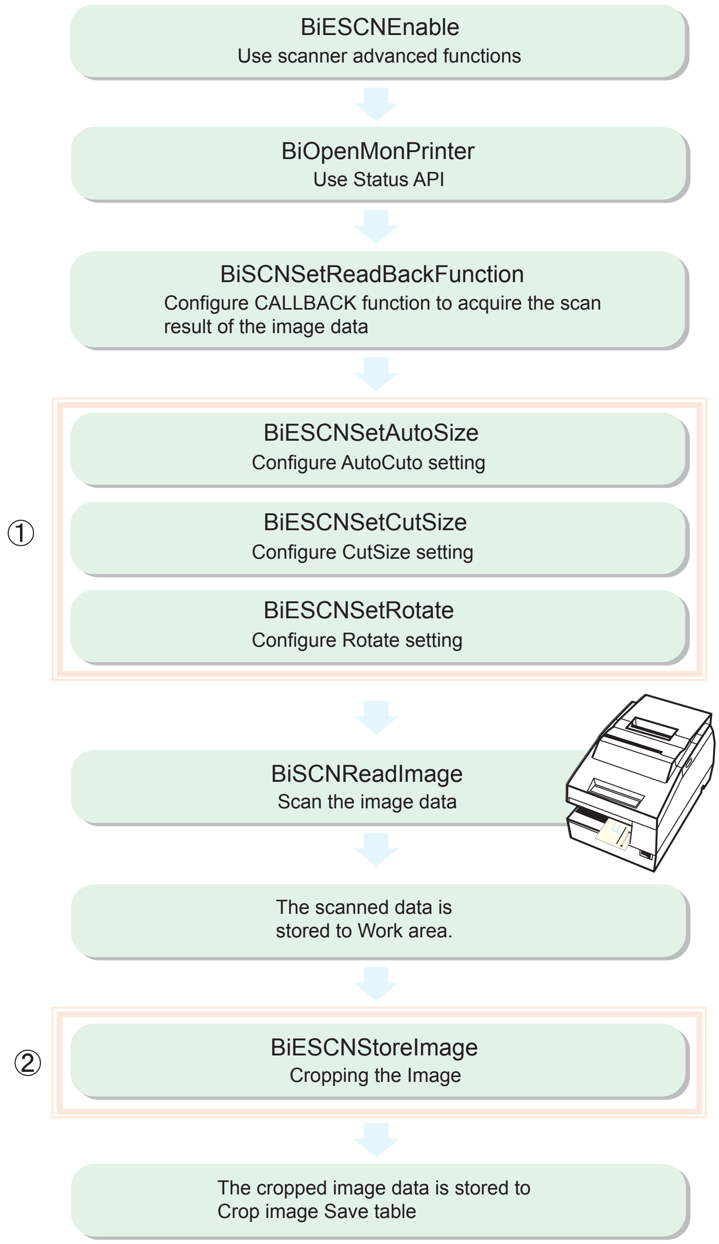
nErr = BiSCNCancelReadBack (nHandle)

Image data process

Process the image data by using the Status API scanner advanced functions.
The image data can be processed as follows.

1 Processing when reading the image data (Configured before calling BiSCNReadImage)

2 Processing after saving the image data to the work area



BiESCNEnable
Use scanner advanced functions

BiOpenMonPrinter
Use Status API

BiSCNSetReadBackFunction
Configure CALLBACK function to acquire the scan result of the image data

①

BiESCNSetAutoSize
Configure AutoCuto setting

BiESCNSetCutSize
Configure CutSize setting

BiESCNSetRotate
Configure Rotate setting

BiSCNReadImage
Scan the image data

The scanned data is stored to Work area.

②

BiESCNStoreImage
Cropping the Image

The cropped image data is stored to Crop image Save table

Start using the scanner advanced functions

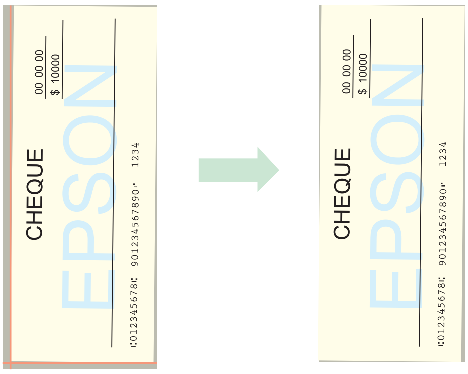
To use the scanner advanced functions, Call BiESCNEnable before calling BiOpenMonPrinter that starts Status API. BiESCNEnable is called as follows. Refer to "BiESCNEnable" on page 193.

nErr = BiESCNEnable (b.StoreType)

Processing when acquiring the image data

Configure the settings to call BiSCNReadImage and process when acquiring the image data. The following functions are available. For details regarding Status API, refer to "Reference for Win32" on page 41.

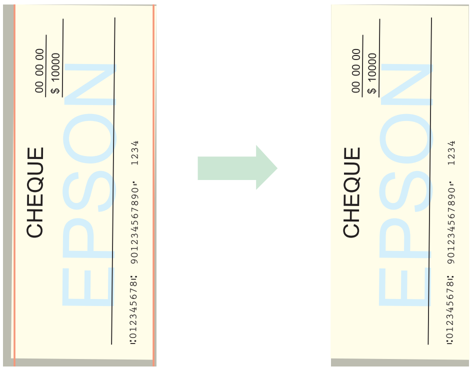
- AutoCuto Process



This process crops the acquired image data by removing the left and bottom black margins.

Status API	Page	Description
BiESCNGetAutoSize	194	Acquiring AutoCuto process settings
BiESCNSetAutoSize	195	Configuring AutoCuto process settings

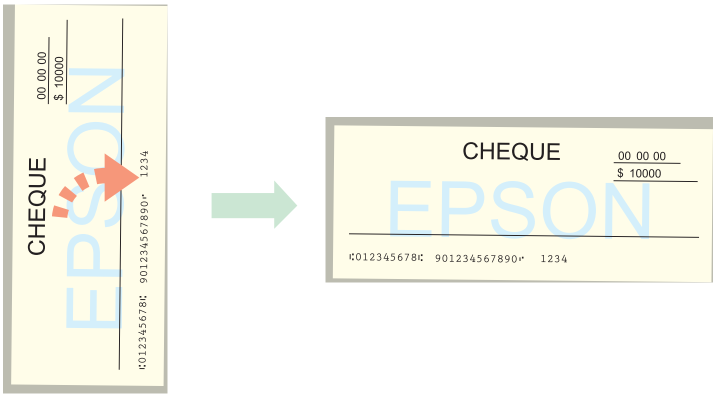
- CutSize Process



This process crops the acquired image data to the specified size by removing the left and right margins.

Status API	Page	Description
BiESCNGetCutSize	196	Acquiring CutSize process settings
BiESCNSetCutSize	197	Configuring CutSize process settings

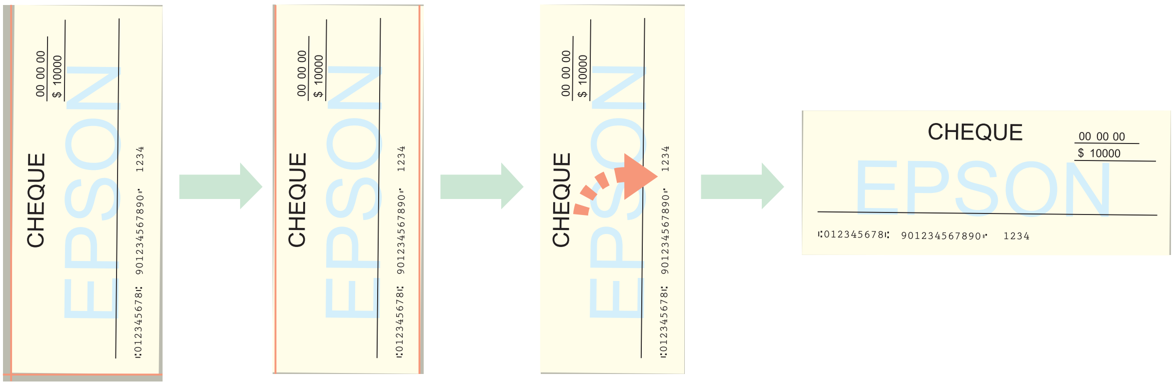
- Rotate Process



This process rotates the acquired image data to the right by 90 degrees.

Status API	Page	Description
BiESCNGetRotate	198	Acquiring Rotate process settings
BiESCNSetRotate	199	Configuring Rotate process settings

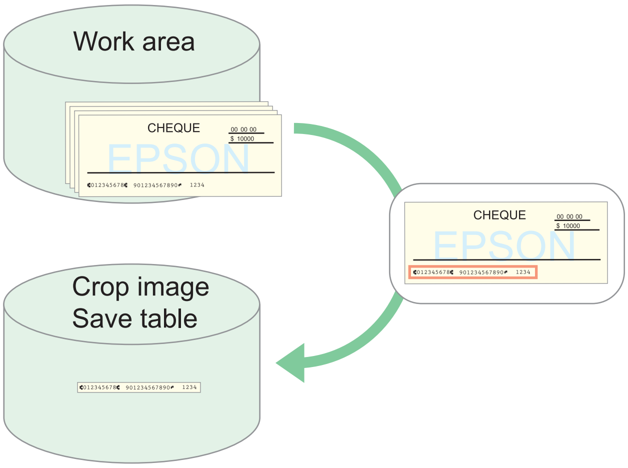
The following image data is processed by combining the AutoCuto process, CutSize process, and the Rotate process.



Processing after saving the image data to the work area

This function processes the image data acquired by BiSCNReadImage after saving the data to the work area.

Crop process



This process crops the specified area of the image data saved in the work area and names and saves the image data. For details regarding Status API, refer to "Reference for Win32" on page 41.

Status API	Page	Description
BiESCNDefineCropArea	202	Registers and deletes the all of the cropped areas
BiESCNGetMaxCropAreas	204	Acquires the number of cropped areas that can be registered
BiESCNStoreImage	205	The registered area of the specified CropArea ID is cropped from the image data in the work area and saved to the Crop image save table.
BiESCNRetrieveImage	208	Acquires cropped image data from the Crop image save table
BiESCNClearImage	211	Deletes cropped image data from the Crop image save table
BiESCNGetRemainingImages	213	Acquires the number of image data that can be saved to the Crop image save table

Reference for Win32

This chapter describes the Status API and syntax used for each model. Refer to " Model Information" regarding the ASB and ink status for each printer, issues causing the printer to go offline, and maintenance counters.

NOTE The data type is described in C++.

Status API used for each model

Status API used for each model is as follows.

BA-T500

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

EU-T300

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

EU-T400

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiCancelError	75	BiGetType	77
BiGetOfflineCode	78	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiSetPresenterEventBackFunction	101
BiSetPresenterEventBackFunctionEx	103	BiSetPresenterEventBackWnd	104
BiCancelPresenterEventBack	105	BiEjectSheet	110
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiPreparePaperChange	119
BiRetractSheet	120	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetReadWaitTimeOut	128		

EU-T500

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiCancelError	75	BiGetType	77
BiGetOfflineCode	78	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiSetPresenterEventBackFunction	101
BiSetPresenterEventBackFunctionEx	103	BiSetPresenterEventBackWnd	104
BiCancelPresenterEventBack	105	BiEjectSheet	110
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiPreparePaperChange	119
BiRetractSheet	120	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetReadWaitTimeOut	128		

TM-C100

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiGetOfflineCode	78
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetInkStatus	88	BiSetInkStatusBackFunction	89
BiSetInkStatusBackFunctionEx	91	BiSetInkStatusBackWnd	93
BiCancelInkStatusBack	94	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetReadWaitTimeOut	128

TM-H5000II

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128
BiMICRCleaning	129	BiMICRGetStatus	130
BiMICRLoadCheck	132	BiMICRReadCheck	133
BiMICREjectCheck	135	BiMICRCancelWaitCheckInsertion	136
BiMICRRetransmissionCheckData	137	BiMICRSetReadBackFunction	141
BiMICRSetReadBackFunctionEx	144	BiMICRSetReadBackWnd	146
BiMICRCancelReadBack	148		

TM-H5200

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128
BiMICRCleaning	129	BiMICRGetStatus	130
BiMICRLoadCheck	132	BiMICRReadCheck	133
BiMICREjectCheck	135	BiMICRCancelWaitCheckInsertion	136
BiMICRRetransmissionCheckData	137	BiMICRSetReadBackFunction	141
BiMICRSetReadBackFunctionEx	144	BiMICRSetReadBackWnd	146
BiMICRCancelReadBack	148		

TM-H6000II

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128
BiMICRCleaning	129	BiMICRGetStatus	130
BiMICRLoadCheck	132	BiMICRReadCheck	133
BiMICREjectCheck	135	BiMICRCancelWaitCheckInsertion	136
BiMICRRetransmissionCheckData	137	BiMICRSelectDataHandling	139
BiMICRSetReadBackFunction	141	BiMICRSetReadBackFunctionEx	144
BiMICRSetReadBackWnd	146	BiMICRCancelReadBack	148
BiSCNClumpPaper	149	BiSCNGetClumpStatus	150
BiSCNGetCroppingArea	152	BiSCNSetCroppingArea	154
BiSCNDeleteCroppingArea	156	BiSCNGetImageFormat	157
BiSCNSetImageFormat	159	BiSCNGetImageList	161
BiSCNGetImageQuality	163	BiSCNSetImageQuality	165
BiSCNGetImageRemainingCapacity	167	BiSCNGetScanArea	169
BiSCNSetScanArea	171	BiSCNPreScan	174
BiSCNReadImage	176	BiSCNRetransmissionImage	179
BiSCNDeleteImage	181	BiSCNSelectScanUnit	183
BiSCNSetReadBackFunction	185	BiSCNSetReadBackFunctionEx	188
BiSCNSetReadBackWnd	190	BiSCNCancelReadBack	192
BiESCNEnable	193	BiESCNGetAutoSize	194
BiESCNSetAutoSize	195	BiESCNGetCutSize	196
BiESCNSetCutSize	197	BiESCNGetRotate	198
BiESCNSetRotate	199	BiESCNGetDocumentSize	200
BiESCNSetDocumentSize	201	BiESCNDefineCropArea	202
BiESCNGetMaxCropAreas	204	BiESCNStoreImage	205
BiESCNRetrieveImage	208	BiESCNClearImage	211
BiESCNGetRemainingImages	213		

TM-H6000III

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128
BiMICRCleaning	129	BiMICRGetStatus	130
BiMICRLoadCheck	132	BiMICRReadCheck	133
BiMICREjectCheck	135	BiMICRCancelWaitCheckInsertion	136
BiMICRRetransmissionCheckData	137	BiMICRSelectDataHandling	139
BiMICRSetReadBackFunction	141	BiMICRSetReadBackFunctionEx	144
BiMICRSetReadBackWnd	146	BiMICRCancelReadBack	148
BiSCNClumpPaper	149	BiSCNGetClumpStatus	150
BiSCNGetCroppingArea	152	BiSCNSetCroppingArea	154
BiSCNDeleteCroppingArea	156	BiSCNGetImageFormat	157
BiSCNSetImageFormat	159	BiSCNGetImageList	161
BiSCNGetImageQuality	163	BiSCNSetImageQuality	165
BiSCNGetImageRemainingCapacity	167	BiSCNGetScanArea	169
BiSCNSetScanArea	171	BiSCNPreScan	174
BiSCNReadImage	176	BiSCNRetransmissionImage	179
BiSCNDeleteImage	181	BiSCNSelectScanUnit	183
BiSCNSetReadBackFunction	185	BiSCNSetReadBackFunctionEx	188
BiSCNSetReadBackWnd	190	BiSCNCancelReadBack	192
BiESCNEnable	193	BiESCNGetAutoSize	194
BiESCNSetAutoSize	195	BiESCNGetCutSize	196
BiESCNSetCutSize	197	BiESCNGetRotate	198
BiESCNSetRotate	199	BiESCNGetDocumentSize	200
BiESCNSetDocumentSize	201	BiESCNDefineCropArea	202
BiESCNGetMaxCropAreas	204	BiESCNStoreImage	205
BiESCNRetrieveImage	208	BiESCNClearImage	211
BiESCNGetRemainingImages	213		

TM-J7000/7100

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetOfflineCode	78
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetInkStatus	88	BiSetInkStatusBackFunction	89
BiSetInkStatusBackFunctionEx	91	BiSetInkStatusBackWnd	93
BiCancelInkStatusBack	94	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128
BiMICRCleaning	129	BiMICRGetStatus	130
BiMICRLoadCheck	132	BiMICRReadCheck	133
BiMICREjectCheck	135	BiMICRCancelWaitCheckInsertion	136
BiMICRRetransmissionCheckData	137	BiMICRSelectDataHandling	139
BiMICRSetReadBackFunction	141	BiMICRSetReadBackFunctionEx	144
BiMICRSetReadBackWnd	146	BiMICRCancelReadBack	148

TM-J7500/7600

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetOfflineCode	78
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetInkStatus	88	BiSetInkStatusBackFunction	89
BiSetInkStatusBackFunctionEx	91	BiSetInkStatusBackWnd	93
BiCancelInkStatusBack	94	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128

TM-L60II

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetReadWaitTimeOut	128

TM-L90

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetOfflineCode	78
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-L90Peeler

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetOfflineCode	78
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-P60

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetBatteryStatus	95
BiSetBatteryStatusBackFunctionEx	97	BiSetBatteryStatusBackWndEx	99
BiCancelBatteryStatusBack	100	BiAutoPowerOffTime	106
BiPowerOff	108	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-P60Peeler

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetBatteryStatus	95
BiSetBatteryStatusBackFunctionEx	97	BiSetBatteryStatusBackWndEx	99
BiCancelBatteryStatusBack	100	BiAutoPowerOffTime	106
BiPowerOff	108	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-T70

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiPowerOff	108
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-T88III

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128

TM-T88IV

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiPowerOff	108
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-T88IV ReStick

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiPowerOff	108
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-T90

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetOfflineCode	78
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetCounter	111	BiResetCounter	113
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-U120

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiCancelError	75	BiGetType	77
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetReadWaitTimeOut	128

TM-U120II

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiCancelError	75	BiGetType	77
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetPrnCapability	115	BiOpenDrawer	117
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetReadWaitTimeOut	128

TM-U220

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128

TM-U230

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-U295

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetReadWaitTimeOut	128

TM-U375

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetReadWaitTimeOut	128

TM-U590

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetPrnCapability	115
BiSendDataFile	121	BiDirectSendRead	123
BiSetDefaultEchoTime	126	BiSetEtherEchoTime	127
BiSetReadWaitTimeOut	128		

TM-U675

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiSetMonEtherInterval	66
BiDirectIO	67	BiDirectIOEx	69
BiResetPrinter	73	BiCancelError	75
BiGetType	77	BiGetStatus	79
BiGetRealStatus	80	BiSetStatusBackFunction	82
BiSetStatusBackFunctionEx	84	BiSetStatusBackWnd	86
BiCancelStatusBack	87	BiGetCounter	111
BiResetCounter	113	BiGetPrnCapability	115
BiOpenDrawer	117	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetEtherEchoTime	127	BiSetReadWaitTimeOut	128
BiMICRCleaning	129	BiMICRGetStatus	130
BiMICRLoadCheck	132	BiMICRReadCheck	133
BiMICREjectCheck	135	BiMICRCancelWaitCheckInsertion	136
BiMICRRetransmissionCheckData	137	BiMICRSetReadBackFunction	141
BiMICRSetReadBackFunctionEx	144	BiMICRSetReadBackWnd	146
BiMICRCancelReadBack	148		

TM-U950

Status API	Page	Status API	Page
BiOpenMonPrinter	59	BiCloseMonPrinter	61
BiLockPrinter	62	BiUnlockPrinter	64
BiSetMonInterval	65	BiDirectIO	67
BiDirectIOEx	69	BiResetPrinter	73
BiCancelError	75	BiGetType	77
BiGetStatus	79	BiGetRealStatus	80
BiSetStatusBackFunction	82	BiSetStatusBackFunctionEx	84
BiSetStatusBackWnd	86	BiCancelStatusBack	87
BiGetPrnCapability	115	BiSendDataFile	121
BiDirectSendRead	123	BiSetDefaultEchoTime	126
BiSetReadWaitTimeOut	128	BiMICRCleaning	129
BiMICRGetStatus	130	BiMICRLoadCheck	132
BiMICRReadCheck	133	BiMICREjectCheck	135
BiMICRCancelWaitCheckInsertion	136	BiMICRRetransmissionCheckData	137
BiMICRSetReadBackFunction	141	BiMICRSetReadBackFunctionEx	144
BiMICRSetReadBackWnd	146	BiMICRCancelReadBack	148

BiOpenMonPrinter

Makes Status API available for the printer and returns the handle.
You can open one printer from multiple processes at the same time.
When you open the opened printer from the same process again, a new different handle will return. In such a case, both handles are valid.

Syntax

nHandle = **BiOpenMonPrinter** *(INT nType, LPSTR pName)*

Example)

- Make Status API available from the port.
  nHandle = BiOpenMonPrinter(1,"ESDPRT001");
- Make Status API available from the printer.
  nHandle = BiOpenMonPrinter(2, "EPSON TM-T88IV Receipt");

Argument

nType: Specifies the pName type. This is an INT type.

Macro Definition (Constant)	Value	Description
TYPE_PORT	1	Specify the port name in *pName*.
TYPE_PRINTER	2	Specify the printer name in *pName*.

pName: If 1 is specified in nType, specify the port name (example: "ESDPRT001").
If 2 is specified, specify the printer name (example: "EPSON TM-T88IV Receipt").
This is a LPSTR type.

Return value

Returns the variable (*nHandle*) defined in INT type. If Status API is successfully used, the handle identifying the printer is returned to *nHandle* (correct value). The handle is returned even if the printer is offline.

The following Status API execution errors (value) are returned.

Macro Definition (Constant)	Value	Description
ERR_TYPE	-10	Parameter error of *nType*
ERR_OPENED	-20	The specified printer is already opened.
ERR_NO_PRINTER	-30	The specified printer driver does not exist
ERR_NO_TARGET	-40	Printer unavailable
ERR_NO_MEMORY	-50	Not enough memory
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer

Macro Definition (Constant)	Value	Description
ERR_PARAM	-90	Parameter error
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_SPL_NOT_EXIST	-350	The spooler service is not operating.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function before using other Status API functions. The handle of the return value is used as the argument by other Status API functions.

NOTE

When this function is called, the specified printer is exclusively available until BiCloseMonPrinter is called. Status API functions from other applications are unavailable during this time.
Acquired handles are only valid within the same application.
The maximum number of printers that can be started at one time is 32.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns the handle to *nHandle*.
Offline	Returns the handle to *nHandle*. However, switch to online as the printer cannot print offline.
Cable Removed/Power Off	Returns "ERR_ACCESS" to *nHandle*.

BiCloseMonPrinter

Cancels the status monitoring printer.

NOTE When a BiOpenMonPrinter function is called, always cancel the status monitoring printer using the BiCloseMonPrinter function. An error is generated if a BiOpenMonPrinter function is called again without canceling.

Syntax

nErr = **BiCloseMonPrinter** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiLockPrinter

Locks the printer.

NOTE

- This API is used for a shared printer.
- When using a local printer, this API is used to control multiple processes.

Syntax

nErr = **BiLockPrinter** (*nHandle, timeout*)

Argument

nHandle: Specifies the handle. This is an INT type.

timeout: Specifies the timeout time in ms (milliseconds). Specify it with a positive value. This is a DWORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

This API allows you to access the TM printer exclusively. The BiUnlockPrinter API is provided for canceling the exclusive access. While the TM printer is exclusively accessed, the printer does not accept any other API requests of direct access to the printer. The printer will return ERR_LOCKED to those other API requests.
The exclusive access right to the TM printer is given to a process. Therefore, the exclusive API access is available from other threads in the same process that is locking the printer.

Executing of API in the same process can be repeated. In this case, the printer is locked with multiple accesses. To unlock the printer, execute BiUnlockPrinter the number of times the API has been executed.

When executing exclusive access from a client to a shared printer or to a local printer via Ethernet, the access status is interrupted if the connection is lost, and restored upon recovery of the connection.
However, while the exclusive access status is being interrupted, another process can lock the TM printer for exclusive access. Once the printer is locked by another process, the printer returns ERR_LOCKED to API of the previous process. When another process is finished unlocking the printer, the exclusive access status of the previous process is restored.

Possible causes of the connection failure are as follows.

[Failure during exclusive access to a printer connected via Ethernet]

- The printer is turned Off, or the Ethernet connection between the computer and the printer is disconnected.
- The computer has entered Standby or Hibernate mode.

[Failure during exclusive access to a shared printer from a client]

- The connection between the client and the server is disconnected.
- The client computer has entered Standby or Hibernate mode.

BiUnlockPrinter

Unlocks the lock of the printer.

NOTE
- This API is used for a shared printer.
- When using a local printer, this API is used to control multiple processes.

Syntax

nErr = **BiUnlockPrinter** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE
For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

This API unlocks the printer locked by “BiLockPrinter”. After the lock is canceled, the printer can accept the API from other processes.

If you execute this API when the printer is not locked, “SUCCESS” will be returned to *Return value*.

BiSetMonInterval

Configures the time interval for Status API to read printer status.

CAUTION Setting the interval to a long time overflows the serial receiving buffer and an accurate ASB status cannot be acquired.

Syntax

nErr = **BiSetMonInterval** (*nHandle, wNoPrnInterval, wPrnInterval*)

Argument

nHandle: Specifies the handle. This is an INT type.

wNoPrnInterval: This is not applicable. This is a WORD type.

wPrnInterval: Configure the status monitoring interval of the printer in ms(millisecond) units. This is a WORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiSetMonEtherInterval

Status API configures the read interval of the network printer status.

Syntax

nErr = **BiSetMonEtherInterval** (*nHandle, wEtherInterval*)

Argument

nHandle: Specifies the handle. This is an INT type.

wEtherInterval: Configure the status monitoring interval of the network printer in ms (millisecond) units. This is a WORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiDirectIO

Sends special commands (ESC/POS command) to the printer. Can also acquire command execution results from the printer. It is recommended that BiDirectIOEx is used to acquire execution results.

> NOTE
>
> Contact the dealer regarding ESC/POS commands.

Syntax

nErr = **BiDirectIO** (*nHandle, writeLen, writeCmd, readLen, readBuff, Timeout, nullTerminate*)

Refer to the next argument.

Argument

nHandle: Specifies the handle. This is an INT type.

writeLen: Specifies the data length to write to the printer. Does not write to the printer when "0".
This is a BYTE type.

writeCmd: Specifies the data (ESC/POS command) to write to the printer. This is a LPBYTE type.

readLen: Specifies the data length read from the printer.
Specify when the command execution results are required from the printer.
Specify as "0" when not required. This is a LPBYTE type.

readBuff: Specifies the buffer saving the data read from the printer.
This is a LPBYTE type.

Timeout: Specifies the timeout time in ms (milliseconds). This is a DWORD type.

nullTerminate: In the case of "True", reading is complete when NULL is received from the printer. At this time, specify the readBuff size to readLen.
In the case of "FALSE ", the length of data specified in readLen is read or data is read from the printer until a timeout error is generated.

> NOTE
>
> Ensure that the size of readBuff is the same length specified in readLen or longer.
> Contact the dealer regarding ESC/POS commands.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Confirm proper execution of the function by confirming the return value of *nErr* or proper command execution by confirming printer operation. If execution results are acquired from the printer (specify *readLen*), confirm the execution results.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to *nErr.* Executes the command.
Offline	When "0" is specified to *readLen*: Returns "SUCCESS" to *nErr.* The command executes after the printer is online. When a value over "1" is specified to *readLen*: Returns "ERR_TIMEOUT" to *nErr.* Does not execute the command.
Cable Removed/Power Off	Returns "ERR_ACCESS" to *nErr.* Does not execute the command.
Printing	Returns "ERR_ACCESS" to *nErr.* Does not execute the command.

BiDirectIOEx

Sends special commands (ESC/POS command) to the printer. Can also acquire command execution results from the printer. The ASB suppress command can be added for differences with BiDirectIO. When the ASB suppress command is added, separate data (ASB status, etc.) is not sent from the printer until this function is complete, therefore, this is recommended when receiving execution results from the printer.

> NOTE
>
> When considering expandability and versatility, it is recommended to use BiDirectIOEx rather than using the BiDirectIO function.
> Contact the dealer regarding ESC/POS commands.

Syntax

nErr = **BiDirectIOEx** (*nHandle, writeLen, writeCmd, readLen, readBuff, Timeout,*
nullTerminate, option)

Refer to the next argument.

Argument

nHandle: Specifies the handle. This is an INT type.

writeLen: Specifies the data length to write to the printer. Does not write to the printer when "0". This is a DWORD type.

writeCmd: Specifies the data (ESC/POS command) to write to the printer. This is a LPBYTE type.

readLen: Specifies the data length read from the printer.
Specify when the command execution results are required from the printer.
Specify as "0" when not required. This is a LPDWORD type.

readBuff: Specifies the buffer saving the data read from the printer.
This is a LPBYTE type.

Timeout: Specifies the timeout time in ms (milliseconds). This is a DWORD type.

nullTerminate: In the case of "True", reading is complete when NULL is received from the printer. At this time, specify the readBuff size to readLen.
In the case of "FALSE", the length of data specified in readLen is read or data is read from the printer until a timeout error is generated.

option: Controls the ASB suppression command. This is a BYTE type.

Value	Description
0	Send an ASB suppression command before writing data and enable ASB after reading the data.
1	Does not send the ASB suppression command or ASB enable command.

NOTE

Ensure that the size of readBuff is the same length specified in readLen or longer.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Not enough memory
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Confirm proper execution of the function by confirming the return value of *nErr* or proper command execution by confirming printer operation. If execution results are acquired from the printer (specify *readLen*), confirm the execution results.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to *nErr.* Executes the command.
Offline	When "0" is specified to *readLen*: Returns "SUCCESS" to *nErr.* The command executes after the printer is online. When a value over "1" is specified to *readLen*: Returns "ERR_TIMEOUT" to *nErr.* Does not execute the command.
Cable Removed/Power Off	Returns "ERR_ACCESS" to *nErr.* Does not execute the command.
Printing	Returns "ERR_ACCESS" to *nErr.* Does not execute the command.

Caution

- Although the maximum data length that can be specified for Read/Write is 2GB, specify the required minimum data length.
- Do not send invalid commands of ASB status transmissions using this function while monitoring the status of the printer. Subsequent status cannot be acquired.
- The ASB (automatic status notification) suppression command ensures that unintended data is not received when sending commands requesting a response from the printer.
  If you do not use the ASB suppression, ensure that the programming considers the reception of unintended data.

- Specifying the receiving buffer processes the data received from the printer using this function or processes the data using the same process as the monitoring sled (BiGetStatus function, etc.). Refer to the following.

Transmission Command	Receiving Buffer Specified	Receiving Buffer	Operation of the Monitoring Sled
Acquiring status command	Yes	Saves the ASB status to the receiving buffer	Does not callback Does not renew the status
	No	-	Does not callback Does not renew the status
Presenter Command	Yes	Saves the presenter response to the receiving buffer	Does not callback
	No	-	Calls back the presenter
Command with responses from other printers	Yes	Enters the printer response into the receiving buffer	Does not effect the monitoring sled
	No	-	Abnormal callbacks may be generated
Command without responses from other printers	Yes	Generates timeout error	Does not effect the monitoring sled
	No	-	Does not effect the monitoring sled

BiResetPrinter

Resets status monitoring printers.

NOTE

- Cancels print jobs when this is called while printing.
- The TM printer with the serial interface cannot be reset.

Syntax

nErr = **BiResetPrinter** (*nHandle*)

Specify the handle in *nHandle*.

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

4

Comment

Confirm proper execution of the function by confirming the return value of *nErr* or by resetting the printer and confirming that the printer is online (confirming the ASB status).

NOTE

After this function is executed, the printer cannot receive a print command for 15 seconds. If print is executed during this time, the job is sent to the spooler and the print processes is executed after the passage of the aforementioned time.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to *nErr* and resets.
Offline	Returns "SUCCESS" to *nErr* and resets.
Cable Removed/ Power Off	Returns "ASB_NO_RESPONSE" to ASB status and does not reset.
Printing	Cancels the print job and resets.

BiCancelError

If a printer recoverable error is generated, execute this function after removing the error cause, and the TM printer recovers from the error.

NOTE

If a printer recoverable error is generated while transmitting data, recovery may not be possible with this function. In this case, use BiResetPrinter after resolving the error cause and recover from the error.

Syntax

nErr = **BiCancelError** (*nHandle*)

Specify the handle in *nHandle*.

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Use BiCancelError as follows.

Error	Response
Cover open error	Call this function after closing the cover.
Auto cutter error	Call this function after removing any paper around the cutter and closing the cover.
Platen open error (EU-T500)	Call this function after closing the platen opened while printing.
BM detection error (EU-T500)	No available black mark is detected. Call this function after loading the paper with black marks in the correct direction.
Presenter error (EU-T500)	A paper jam error is generated in the presenter. Call this function after removing the paper in the presenter.
Journal printing function error (EU-T500)	A calling error of BiCheckPrint in the journal unit is generated. Call this function after removing the paper in the presenter.

BiGetType

Acquires the type ID of the printer.

NOTE For information on the type ID that can be acquired, ask your dealer.

Syntax

nErr = **BiGetType** (*nHandle, typeID, font, exrom, special*)

If you specify a handle to *nHandle*, a type ID is set to *typeID*, Device font is set to *font*. A special ID of the printer returns to *special*.

Argument

nHandle:	Specifies the handle. This is an INT type.
typeID:	A type ID of the printer will be set. This is a LPBYTE type.
font:	Device font will be set. This is a LPBYTE type.
exrom:	This is not applicable. This is a LPBYTE type.
special:	A special ID of the printer will be set. This is a LPBYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiGetOfflineCode

Acquires the cause of the printer to go offline.

NOTE

- While the Ethernet is connected, acquiring the offline cause is not supported.
- If this function is called while the printer is online, the offline cause cannot be acquired. In addition, the cause cannot be acquired when the printer power is OFF or cables have been unplugged.
- Call this function after 500ms have passed after calling BiResetPrinter (printer reset).

Syntax

nErr = **BiGetOfflineCode** (*nHandle, offlinecode*)

Argument

nHandle: Specifies the handle. This is an INT type.

offlinecode: Returns the offline cause as a 5 byte value. This is a LPBYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "Model Information" on page 261 regarding the causes for the printer to go offline.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to nErr and acquires the online value.
Offline	Returns "SUCCESS" to nErr and acquires the offline cause.
Cable Removed/Power Off	Returns "SUCCESS" to nErr but cannot acquire the offline cause.

BiGetStatus

Acquires the current printer status (ASB status).

Syntax

nErr = **BiGetStatus** (*nHandle* , *lpStatus*)

Specify the handle in *nHandle*. Returns ASB status to *lpStatus*.

Argument

nHandle: Specifies the handle. This is an INT type.

lpStatus: Returns the ASB status saved to Status API. This is a LPDWORD type.
The ASB status is a 4 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "Model Information" on page 261 regarding the ASB status that can be acquired by each printer.

NOTE

There are invalid ASB statuses according to the printer model in use.

BiGetRealStatus

Acquires the current printer status (ASB status).

Syntax

nErr = **BiGetRealStatus** (*nHandle* , *lpStatus*)

Argument

nHandle: Specifies the handle. This is an INT type.

lpStatus: Returns the ASB status saved to Status API. This is a LPDWORD type.
The ASB status is a 4 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

This function sends the command to acquire the ASB status to the printer and receives the acquired status after the function is called. That is why, even when printing has been completed, ASB_PRINTSUCCESS is not acquired. Also, when the power is turned off, ASB_NO_RESPONSE is not acquired because ERR_ACCESS is returned.

Refer to "Model Information" on page 261 regarding the ASB status that can be acquired by each printer.

NOTE

There are invalid ASB statuses according to the printer model in use.

BiSetStatusBackFunction

Automatically acquires the printer status (ASB status) using the callback function when the printer status changes.

NOTE This is unavailable when the development environment is VB.

Syntax

nErr = **BiSetStatusBackFunction** (*nHandle, int (CALLBACK EXPORT *pStatusCB)*
(DWORD dwStatus))

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pStatusCB)(DWORD dwStatus):
Specifies the definition address of the callback function.

dwStatus: Returns the ASB status saved to Status API. This is a DWORD type.
The ASB status is a 4 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function to set the printer status to *dwStatus* and call the callback function. When the printer status changes, new information is automatically set to *dwStatus* and calls the callback function. Cancel this function using BiCancelStatusBack.
Refer to "Model Information" on page 261 regarding the ASB status that can be acquired by each printer.

NOTE

There are invalid ASB statuses according to the printer model in use.
Status API cannot be used within the registered callback function.

BiSetStatusBackFunctionEx

Automatically acquires the printer status (ASB status) using the callback function when the printer status changes.
Identifies the printer port originating the callback, in addition to the functions of BiSetStatusBackFunction.

NOTE This is unavailable when the development environment is VB.

Syntax

nErr = **BiSetStatusBackFunctionEx** (*nHandle, int (CALLBACK EXPORT *pStatusCB)
(DWORD dwStatus, LPSTR lpcPortName)*)

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pStatusCB)(DWORD dwStatus, LPSTR lpcPortName):
Specifies the definition address of the callback function.

dwStatus: Returns the ASB status saved to Status API. The ASB status is a 4 byte configuration. This is a DWORD type.

lpcPortName: Returns the printer port name originating the callback. This is a LPSTR type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function to set the printer status to *dwStatus* and call the callback function. When the printer status changes, new information is automatically set to *dwStatus* and calls the callback function. Cancel this function using BiCancelStatusBack.
Refer to "Model Information" on page 261 regarding the ASB status that can be acquired by each printer.

NOTE	There are invalid ASB statuses according to the printer model in use. Status API cannot be used within the registered callback function.

BiSetStatusBackWnd

Automatically generates a click event and acquires the printer status (ASB status) when the printer status changes.

Syntax

nErr = **BiSetStatusBackWnd** (*nHandle, hWnd, lpStatus*)

Specify the handle in *nHandle*. Returns ASB status to *lpStatus*.

Argument

nHandle: Specifies the handle. This is an INT type.

hWnd: Specifies the window handle of the button generating the click event. This is a Long type.

lpStatus: Returns the ASB status saved to Status API. This is a LPDWORD type. The ASB status is a 4 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "Model Information" on page 261 regarding the ASB status that can be acquired by each printer.

NOTE

There are invalid ASB statuses according to the printer model in use.
Status API cannot be used from the specified window handle.

BiCancelStatusBack

Cancels the automatic status notification request process called using the BiSetStatusBackFunction, BiSetStatusBackFunctionEx, or BiSetStatusBackWnd function.

Syntax

nErr = **BiCancelStatusBack** (*nHandle*)

Specify the handle in *nHandle*.

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

Returns "SUCCESS" even when executed when the automatic status notification request process is not registered. For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiGetInkStatus

Acquires the current ink status.

Syntax

nErr = **BiGetInkStatus** (*nHandle, lpStatus*)

Argument

nHandle: Specifies the handle. This is an INT type.

lpStatus: The current ink status is returned in a LPWORD type.
The ink status is a 2 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "Model Information" on page 261 regarding the ink status that can be acquired by each printer.

BiSetInkStatusBackFunction

Automatically acquires the ink status using the callback function when the ink status changes.

NOTE This is unavailable when the development environment is VB 6.0.

Syntax

nErr = **BiSetInkStatusBackFunction** (*int nHandle, int (CALLBACK EXPORT *pStatusCB) (WORD wStatus)*)

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pStatusCB)(WORD wStatus):
Specifies the definition address of the callback function.

wStatus: Automatically returns the ink status. This is a WORD type.
The ink status is a 2 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function to set the printer status to *wStatus* and call the callback function. When the ink status changes, new information is automatically set to *wStatus* and the callback function is called. Cancel this function using BiCancelInkStatusBack.
Refer to "Model Information" on page 261 regarding the ink status that can be acquired by each printer.

NOTE

Status API cannot be used within the registered callback function.

BiSetInkStatusBackFunctionEx

Automatically acquires the ink status using the callback function when the ink status changes. Identifies the printer port originating the callback, in addition to the functions of BiSetInkStatusBackFunction.

> NOTE
> This is unavailable when the development environment is VB 6.0.

Syntax

nErr = **BiSetInkStatusBackFunctionEx** (*nHandle, int (CALLBACK EXPORT *pStatusCB)*
(WORD wStatus, LPSTR lpcPortName))

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pStatusCB)(WORD wStatus, LPSTR lpcPortName):
Specifies the definition address of the callback function.

wStatus: Automatically returns the ink status. This is a WORD type.
The ink status is a 2 byte configuration.

lpcPortName: Returns the printer port name originating the callback. This is a LPSTR type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

> NOTE
> For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function to set the ink status to *wStatus* and call the callback function. When the printer status changes, new information is automatically set to *wStatus* and the port name of the acquired destination is set to *lpcPortName* and the callback function is called. Cancel this function using BiCancelInkStatusBack.

Refer to "Model Information" on page 261 regarding the ink status that can be acquired by each printer.

NOTE

Status API cannot be used within the registered callback function.

BiSetInkStatusBackWnd

Automatically acquires the ink status generating a click event when the ink status changes.

Syntax

nErr = **BiSetInkStatusBackWnd** (*nHandle, hWnd, lpStatus*)

Argument

nHandle: Specifies the handle. This is an INT type.

hWnd: Specifies the window handle of the button generating the click event.
This is a Long type.

lpStatus: Automatically returns the ink status. This is a LPWORD type.
The ink status is a 2 byte configuration.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "Model Information" on page 261 regarding the ink status that can be acquired by each printer.

NOTE

Status API cannot be used from the specified window handle.

BiCancelInkStatusBack

Cancels the automatic ink status notification request process called using the BiSetInkStatusBackFunction, BiSetInkStatusBackFunctionEx, or BiSetInkStatusBackWnd.

Syntax

nErr = **BiCancelInkStatusBack** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

Returns "SUCCESS" even when called while the automatic ink status notification request process is not registered. For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiGetBatteryStatus

Acquires the printer's current battery status.

Syntax

nErr = **BiGetBatteryStatus** (*nHandle* , *lpbPowerStatus* , *lpbBatteryStatus*)

Argument

nHandle:	Specifies the handle. This is an INT type.
lpbPowerStatus:	Specifies the memory address where the power status is set. This is a LPBYTE type.
lpbBatteryStatus:	Specifies the memory address where the remaining amount of the battery is set. This is a LPBYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Printer's power status is set for *lpbPowerStatus*.Printer's battery status is set for *lpbBatteryStatus*.For the power status and the battery status value, refer to Power status and Battery status below.

Power status

Value	Description
30h	Power supply: AC adapter
31h	Power supply: Battery

Battery status

Value	Description
30h	Remaining battery amount: Level H
31h	Remaining battery amount: Level M
32h	Remaining battery amount: Level L
33h	Remaining battery amount: Level S
34h	Battery is not installed.

CAUTION

If the AC adapter is plugged or unplugged while the TM-P60 is printing, an incorrect power status or battery status may be returned. Do not plug or unplug the AC adapter while the TM-P60 is printig. (Restriction)

BiSetBatteryStatusBackFunctionEx

Automatically acquires the printer's current battery status using the callback function when the printer's current battery status changes. Returns the pointer to the port name to recognize from which port the callback is.

NOTE This is unavailable when the development environment is VB 6.0.

Syntax

nErr = **BiSetBatteryStatusBackFunctionEx** (*nHandle, int (CALLBACK EXPORT *pBatteryCB) (BYTE bPowerStatus, BYTE bBatteryStatus, LPSTR lpcPortName)*)

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pBatteryCB) (BYTE bPowerStatus, BYTE bBatteryStatus, LPSTR lpcPortName):
Specifies the definition address of the callback function.

bPowerStatus: Sets the value of the power status for lpbPowerStatus, and then sends the event. This is a BYTE type.

bBatteryStatus: Sets the battery status value for lpbBatteryStatus, and then sends the event. This is a BYTE type.

lpcPortName: Returns the printer port name originating the callback. This is a LPSTR type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "BiGetBatteryStatus" on page 95 regarding the printer's current battery status that can be acquired by each printer.

BiSetBatteryStatusBackWndEx

Automatically acquires the printer's current battery status generating a click event when the printer's current battery status changes, and the pointer to the port name.

Syntax

nErr = **BiSetBatteryStatusBackWndEx** (*nHandle* , *hWnd* , *lpbPowerStatus* , *lpbBatteryStatus* , *lpcPortName*)

Argument

nHandle: Specifies the handle. This is an INT type.

hWnd: Specifies the window handle of the button generating the click event. This is a Long type.

lpbPowerStatus: Sets the value of the power status for lpbPowerStatus, and then sends the event. This is a LPBYTE type.

lpbBatteryStatus: Sets the battery status value for lpbBatteryStatus, and then sends the event. This is a LPBYTE type.

lpcPortName: Copies the port name for lpcPortName, and then sends the event. LpcPortName must have enough memory to copy the port name including the end NULL. This is a LPSTR type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "BiGetBatteryStatus" on page 95 regarding the printer's current battery status that can be acquired by each printer.

BiCancelBatteryStatusBack

Cancels the battery status notification request registered using BiSetBatteryStatusBackFunctionEx/BiSetBatteryStatusBackWndEx.

Syntax

nErr = **BiCancelBatteryStatusBack** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiSetPresenterEventBackFunction

Automatically acquires the operating results of the presenter using the callback function.

NOTE This is unavailable when the development environment is VB.

Syntax

nErr = **BiSetPresenterEventBackFunction** (*nHandle, int* (CALLBACK EXPORT *pEventCB)
(BYTE returncode))

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pEventCB)(BYTE returncode):
Specifies the definition address of the callback function.

returncode: Returns the operating results of the presenter. This is a BYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function to set the operating results of the presenter to returncode and call the callback function. Cancel this function using BiCancelPresenterEventBack.
The operating results of the acquiring presenter are as follows.

returncode	Description
20h	Paper removed
21h	Retract successful
22h	Retract failed
23h	Eject successful
24h	Eject failed
25h	No paper
26h	Roll paper exchange preparation successful
27h	Roll paper exchange preparation failed
41h	Timeout retract successful
42h	Timeout retract failed
43h	Timeout eject successful
44h	Timeout eject failed

BiSetPresenterEventBackFunctionEx

Automatically acquires the operating results of the presenter using the callback function. Identifies the printer port originating the callback, in addition to the functions of BiSetPresenterEventBackFunction.

NOTE This is unavailable when the development environment is VB.

Syntax

nErr = **BiSetPresenterEventBackFunctionEx** (*nHandle, int* (CALLBACK EXPORT *pEventCB)
(BYTE returncode,
LPSTR lpcPortName))

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pEventCB)(BYTE returncode, LPSTR lpcPortName):
Specifies the definition address of the callback function.

returncode: Returns the operating results of the presenter. This is a BYTE type.

lpcPortName: Returns the printer port name originating the callback. This is an LPSTR type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call this function to set the operating results of the presenter to returncode and call the callback function. Cancel this function using BiCancelPresenterEventBack.

Refer to "BiSetPresenterEventBackFunction" on page 101.

BiSetPresenterEventBackWnd

Automatically generates a click event and acquires the operating results of the presenter.

Syntax

nErr = **BiSetPresenterEventBackWnd** (*int nHandle, long hWnd, LPBYTE returncode*)

Argument

nHandle: Specifies the handle. This is an INT type.

hWnd Specifies the window handle of the button generating the click event. This is a Long type.

returncode: Returns the operating results of the presenter. This is a LPBYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Refer to "BiSetPresenterEventBackFunction" on page 101 for the operating results of the acquiring presenter. Cancel this function using BiCancelPresenterEventBack.

BiCancelPresenterEventBack

Cancels the automatic status notification request process of the presenter called by the BiSetPresenterEventBackFunction, BiSetPresenterEventBackFunctionEx, or BiSetPresenterEventBackWnd function.

Syntax

nErr = **BiCancelPresenterEventBack** (*int nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiAutoPowerOffTime

Updates the maintenance counter and prepares to turn off the power to the printer.
Cannot turn off the power to the printer.

NOTE Cannot call when in online recovery standby.

Syntax

nErr = **BiAutoPowerOffTime** (*nHandle , bMode , lpbTime*)

Argument

nHandle: Specifies the handle. This is an INT type.

bMode: Specifies the process mode. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_GET	0	Acquires the auto power off time.
EPS_BI_SET	1	Sets the auto power off time.

lpbTime: Specifies the memory address to set the auto power off time.
This is a LPBYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Not enough memory
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Sets the auto power off time.

lpbTime	Auto power off time
0	Auto power off: disabled
1 - 60	1 to 60 minutes

BiPowerOff

Updates the maintenance counter and prepares to turn off the power to the printer.
Cannot turn off the power to the printer.

NOTE Cannot call when in online recovery standby.

Syntax

nErr = **BiPowerOff** (*nHandle*)

Specify the handle in *nHandle*.

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Not enough memory
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

The printer operation performed by this API differs with the TM printer model.

Operation	TM-P60	TM-P60Peeler	TM-T70	TM-T88IV
Stores the maintenance counter value.	√	√	√	√
Places the interface in BUSY state.	√	√	√	√
Places the TM printer in standby mode with power off.	√	√	√	√
Turns the printer power off.	√	√	-	-

BiEjectSheet

Ejects the presenter's receipt paper.

Syntax

nErr = **BiEjectSheet** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call BiSetPresenterEventBackFunction/BiSetPresenterEventBackFunctionEx/BiSetPresenterEventBackWnd to confirm the successful execution of the function.

BiGetCounter

Acquires the maintenance counter value.

NOTE

- For information on the counter number and the maintenance counters that can be acquired, ask your dealer.
- The maintenance counter may not be available according to the printer. In this case, a timeout error is generated.
- Confirm that the ASB status is online before calling this function.

Syntax

nErr = **BiGetCounter** (*nHandle, readno, readcounter*)

Specify the handle in *nHandle*. *readno* specifies the acquired maintenance counter number and the maintenance counter value is returned to *readcounter*.

Argument

nHandle: Specifies the handle. This is an INT type.

readno: Specifies the acquired maintenance counter number. This is a WORD type.

readcounter: Returns the maintenance counter. This is a LPDWORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

There are two types of maintenance counters; those that can be reset by the user, and integrated counters that cannot be reset.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to *nErr.* Acquires the maintenance counter value.
Offline	Returns "ERR_TIMEOUT" to *nErr.* Does not acquire the maintenance counter value.
Cable Removed/ Power Off	Returns "ERR_ACCESS" to *nErr.* Does not acquire the maintenance counter value.
Printing	Returns "ERR_ACCESS" to *nErr.* Does not acquire the maintenance counter value.

BiResetCounter

Resets the maintenance counter.

NOTE

- For information on the counter number and the maintenance counters that can be acquired, ask your dealer.
- The maintenance counter may not be available according to the printer. In this case, a timeout error is generated.
- Confirm that the ASB status is online before calling this function.

Syntax

nErr = **BiResetCounter** (*nHandle, readno*)

Specify the handle in *nHandle*. Specify the maintenance counter number reset to *readno*.

Argument

nHandle: Specifies the handle. This is an INT type.

readno: Specifies the maintenance counter number to be reset. This is a WORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

If "SUCCESS" is the return value of *nErr* and the value acquired by BiGetCounter after resetting the maintenance counter, this confirms normal execution.

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to *nErr*. Resets the maintenance counter.
Offline	Returns "ERR_TIMEOUT" to *nErr*. Does not reset the maintenance counter.
Cable Removed/Power Off	Returns "ERR_ACCESS" to *nErr*. Does not reset the maintenance counter.
Printing	Returns "ERR_ACCESS" to *nErr*. Does not reset the maintenance counter.

BiGetPrnCapability

Acquires the specified printer information in printer ID.

NOTE

- For information on the Printer Capability that can be acquired, ask your dealer.
- There are printer ID's that are not supported by printers. If unsupported printer ID's are specified, a timeout error is generated.

Syntax

nErr = **BiGetPrnCapability** (*nHandle, prnID, pBuffSize, pBuff*)

Specify the handle in *nHandle* and specify the acquiring printer information to *prnID*. Specify the memory size to set the printer information in *pBuffSize* and specify the memory address to set the printer information in *pBuff*.

Argument

nHandle: Specifies the handle. This is an INT type.

prnID: Specifies the acquiring printer information. This is a BYTE type.

pBuffSize: Specifies the memory size to set the printer information (1 to 80). Returns the actual read data size after calling this function. In the case of insufficient buffer capacity, the required byte size is returned. This is a LPBYTE type.

pBuff: Specifies the memory address to set the printer information. This is a LPBYTE type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_BUFFER_OVER_FLOW	-140	Lack of buffer capacity.
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.

Macro Definition (Constant)	Value	Description
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiOpenDrawer

Opens the drawer.

> NOTE
>
> Opens the drawer even when the printer is offline.

Syntax

nErr = **BiOpenDrawer** (*nHandle, drawer, pulse*)

Specify the handle in *nHandle*. Specify the drawer to open in *drawer* and specify the time until the drawer opens in *pulse*.

Argument

nHandle: Specifies the handle. This is an INT type.

drawer: Specifies the drawer to open. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_DRAWER_1	1	Opens drawer 1
EPS_BI_DRAWER_2	2	Opens drawer 2

pulse: Specifies the time until the drawer is opened. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_PLUSE_100	1	Operates the drawer after 100 milliseconds
EPS_BI_PLUSE_200	2	Operates the drawer after 200 milliseconds
EPS_BI_PLUSE_300	3	Operates the drawer after 300 milliseconds
EPS_BI_PLUSE_400	4	Operates the drawer after 400 milliseconds
EPS_BI_PLUSE_500	5	Operates the drawer after 500 milliseconds
EPS_BI_PLUSE_600	6	Operates the drawer after 600 milliseconds
EPS_BI_PLUSE_700	7	Operates the drawer after 700 milliseconds
EPS_BI_PLUSE_800	8	Operates the drawer after 800 milliseconds

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

The following operations are executed according to the printer status when this function is called.

Printer Status	Operation
Online	Returns "SUCCESS" to *nErr.* Opens the drawer.
Offline	Returns "SUCCESS" to *nErr.* Opens the drawer.
Cable Removed/Power Off	Returns "ERR_ACCESS" to *nErr.* Does not open the drawer.

BiPreparePaperChange

Ejects backwards to prepare for roll paper exchange.

Syntax

nErr = **BiPreparePaperChange** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiRetractSheet

Retracts the receipt paper.

Syntax

nErr = **BiRetractSheet** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Call BiSetPresenterEventBackFunction/BiSetPresenterEventBackFunctionEx/ BiSetPresenterEventBackWnd to confirm the successful execution of the function.

BiSendDataFile

Specify the command definition file to define the transmission command (ESC/POS command).

> NOTE
>
> Specify the command definition file in the specified format.
> Contact the dealer regarding ESC/POS commands.

Syntax

nErr = **BiSendDataFile** (*nHandle, lpcFileName*)

Specify the handle in *nHandle* and the command definition file name in *lpcFileName*.

Argument

nHandle: Specifies the handle. This is an INT type.

lpcFileName: Specifies the command definition file name. This is a LPCSTR type.

Return value

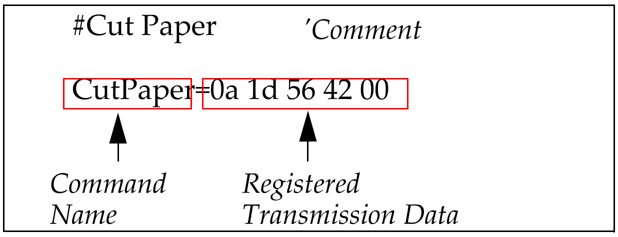
Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Not enough memory
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use

> NOTE
>
> For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

Describe the command definition file using the following format.



```
#Cut Paper          'Comment

CutPaper=0a 1d 56 42 00
```

> NOTE
>
> Contact the dealer regarding ESC/POS commands.

Caution

- Handle character strings following "#" as comments.
- The character string listed to the left of "=" is the "command name" of the data to be written to the printer and the character string listed to the right of "=" is the "registered transmission data".
- Ensure that character strings are listed using parentheses (" ").
- List binary data as two digit hexadecimals.
- The maximum size of "command names" is 33 bytes (33 characters in ANK).
- The maximum size of "registered transmission data" is 10,240 bytes. However, the size of the "registered transmission data" is not the length of an "ASCIIZ character string" but the size after converting the data to binary. Refer to the following examples.

  Example) ABC="ABC": The maximum size of "registered transmission data" is 3 bytes.
  ABC="ABC" 0D 0A: The maximum size of "registered transmission data" is 5 bytes.
  ABC=41 42 43 0D 0A: The maximum size of "registered transmission data" is 5 bytes.
- If a command name is already registered, stop the command registration process and return an error.
- The number of commands that can be registered is limited to the usable memory of the system.
- Call the BiCloseMonPrinter function to cancel the registered command data.

BiDirectSendRead

Executes the command defined in BiSendDataFile(ESC/POScommands).

Syntax

nErr = **BiDirectSendRead** (*nHandle, lpcCmdName, lpcReadName, readLen, pReadBuf, Timeout, nullTerminate*)

Refer to the next argument.

Argument

nHandle: Specifies the handle. This is an INT type.

lpcCmdName: Specifies the "command name" of the command definition file name. This is a LPCSTR type.

lpcReadName: Specifies the "receiving data type name" of the data read from the printer. Refer to Page "BiSendDataFile" on page 121. This is a LPCSTR type.

readLen: Specifies the data length read from the printer. Does not write to the printer when "0". Returns the data length when reading. This is a LPDWORD type.

preadBuff: Specifies the buffer saving the data read from the printer.
This is a LPBYTE type.

Timeout: Specifies the timeout time in ms (milliseconds). This is a DWORD type.

nullTerminate: In the case of "TRUE", reading is complete when NULL is received from the printer. At this time, specify the readBuff size to readLen.
In the case of "FALSE ", the length of data specified in readLen is read or data is read from the printer until a timeout error is generated.

NOTE

Ensure that the size of preadBuff is the same length specified in readLen or longer.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_NO_MEMORY	-50	Not enough memory
ERR_HANDLE	-60	Specified handle is invalid
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_PARAM	-90	Parameter error

Macro Definition (Constant)	Value	Description
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

This function specifies the name (macro name) previously specified in the command definition file. The following data types can be specified as responses from the printer.

Data Types	Description
ASB	Automatic status transmission
ASB Extended	Automatic status transmission regarding extended status
Ptr Info Byte	Printer ID information
Ptr Info String	Printer information B
Power OFF	Power OFF notification
Power ON	Power ON notification
Realtime	Realtime transmission of status
Buffer Clear	Buffer clear
Slip Remaining	Dot count transmission of remaining print area of a single-cut sheet
NVM Image Size	Transmission of full capacity of NV graphics area
NVM Image Free	Transmission of remaining capacity of NV graphics area
NVM Image Keys	Transmission of the key code list of defined NV graphics
NVM Image List	Transmission of the data ID list of image reading results saved to NV memory for storage
NVM User Used	Transmission of usage capacity (byte count of used area)
NVM User Free	Transmission of remaining capacity (byte count of unused area)
NVM User Get	Transmission of the storage data of a specified record
NVM User Keys	Transmission of the key code list of a storage record
NVM Set Mode	Transmission of transfer notification to user settings mode
NVM Get Mswitch	Transmission of memory switch values
NVM Set Size	Transmission of customized values
ASB Ink	Automatic status transmission regarding ink
Ptr Info Type A	Printer information A

Data Types	Description
Test Print	Execution of test print
RAM Image Free	Transmission of remaining area of download graphics area
RAM Image Keys	Transmission of the key code list of defined download graphics
OfflineCode Bit	Transmission of offline response (bit format)
OfflineCode Data	Transmission of offline response (data format)
ProcessID	Transmission of process ID response
Data Types	Description
Buffer Clear24	Buffer clear 24
Other	Data not applicable to the aforementioned

NOTE

Contact the dealer regarding ESC/POS commands.

BiSetDefaultEchoTime

Configures the response confirmation frequency of the network printer and the initial time value for a single timeout.

NOTE Can only be used when connected by Ethernet.

Syntax

nErr = **BiSetDefaultEchoTime** (*Count, Timeout*)

Configures the response confirmation frequency to *Count* and the single timeout time to *Timeout*.

Argument

Count: Configures the response confirmation frequency (1 to 255). This is a BYTE type.

Timeout: Configures the single timeout time (1 to 65535) in ms (millisecond) units. This is a WORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_PARAM	-90	Parameter error
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.

NOTE

- For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.
- When a shred printer is accessed from a client that does not have PortHandler, the printer returns ERR_PH_NOT_EXIST.

Comment

The response confirmation frequency is once and the timeout time is 1 second immediately after installing Status API. The configured value is valid the next time the BiOpenMonPrinter function is called.

BiSetEtherEchoTime

Configures the response confirmation frequency of the network printer and the timeout time for one time after Status API is available.

NOTE Can only be used when connected by Ethernet.

Syntax

nErr = **BiSetEtherEchoTime** (*nHandle , Count, Timeout*)

Specify the handle in *nHandle*. Configures the response confirmation frequency to *Count* and the single timeout time to *Timeout*.

Argument

nHandle: Specifies the handle. This is an INT type.

Count: Configures the response confirmation frequency (1 to 255). This is a BYTE type.

Timeout: Configures the single timeout time (1 to 65535) in ms (millisecond) units.
This is a WORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success
ERR_HANDLE	-60	Specified handle is invalid
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot call as another Status API is in use
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

If a value is not configured to this function, the value set in the BiSetDefaultEchoTime function is used.

BiSetReadWaitTimeOut

This is a compatible API. This API itself has no function.

Syntax

nErr = **BiSetReadWaitTimeOut** (*nHandle* , *wTimeOut*)

Argument

nHandle:	Specifies the handle. This is an INT type.
wTimeOut:	Not used. This is a WORD type.

Return value

Returns the following Status API execution errors (value) to the variable (nErr) defined by the INT type. Returns "SUCCESS" (macro definition) when this function is successfully called.

Macro Definition (Constant)	Value	Description
SUCCESS	0	Success

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiMICRCleaning

Cleans the MICR mechanism.

Syntax

nErr = **BiMICRCleaning** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	This function is unavailable as Status API is used by other applications.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

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. After cleaning is finished, the MICR function is deselected and roll paper is used for the printing sheets.

BiMICRGetStatus

Acquires the MICR status.

Syntax

nErr = **BiMICRGetStatus** (*nHandle* , *pStatus*)

Argument

nHandle: Specifies the handle. This is an INT type.

pStatus: Specifies the memory address where the MICR status is set. See the following MICR states concerning the types of MICR status that are returned. This is a LPBYTE type.

Bit	ON/OFF	Value	Status
0	-	0x00	Fixed to 0
1	-	0x02	Fixed to 1
2	ON	0x04	Not selected
	OFF	0x00	Selected
3	ON	0x08	Wait for insertion
	OFF	0x00	Do not wait for insertion
4	-	0x10	Fixed to 1
5	ON	0x20	No paper in the TOF sensor
	OFF	0x00	Paper in the TOF sensor
6	ON	0x40	No paper in the BOF sensor
	OFF	0x00	Paper in the BOF sensor
7	-	0x00	Fixed to 0

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printerÅfs power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

Macro Definition (Constant)	Value	Description
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiMICRLoadCheck

Loads the check to the check print start position when the MICR function is selected.

Syntax

nErr = **BiMICRLoadCheck** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting.
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiMICRReadCheck

Executes check reading. If 0 is specified for the check insertion wait time, the printer's wait time is canceled. Furthermore, the check insertion wait time after MICR reading is finished is reset to 0, the printer's default value.

Syntax

nErr = **BiMICRReadCheck**(*nHandle* , *readFont* , *waitInsertionTime*)

Argument

nHandle: Specifies the handle. This is an INT type.

readFont: Specifies the reading font. This is a BYTE type.

Value	Description
0	E13B
1	CMC7

waitInsertionTime: Specifies the check insertion wait time.
0 ~ 15 (x 60 sec.) (The printer's default is 0)
This is a BYTE type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_WITHOUT_CB	-130	Cannot execute because either BiSCNSetReadBackFunction or BiSCNSetReadBackWnd has not been called.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting.

Macro Definition (Constant)	Value	Description
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

Be sure to call either BiMICRSetReadBackFunction or BiMICRSetReadBackWnd before calling this function.

BiMICREjectCheck

Ejects the check when the MICR function is selected.

Syntax

nErr = **BiMICREjectCheck** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiMICRCancelWaitCheckInsertion

Cancels check insertion wait.

Syntax

nErr = **BiMICRCancelWaitCheckInsertion** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

Even if called with a printer which does not have the MICR function built in, the slip insertion wait setting is canceled.

BiMICRRetransmissionCheckData

Resends the check reading results.

Syntax

nErr = **BiMICRRetransmissionCheckData** (*nHandle , pReadBuffSize , readCharBuff , pStatus ,pDetail, timeout)*

Argument

nHandle: Specifies the handle. This is an INT type.

pReadBuffSize: Specifies the size of memory where the reading data are set. After this function is executed, the size of the actual reading data is set. This is a LPBYTE type.

readCharBuff: Specifies the memory addresses where the reading data are set. This is a LPBYTE type.

pStatus: Specifies a 1-byte memory address where the reading status is set. See the MICR reading status in BiMICRSetReadBackFunction concerning the status that is returned. This is a LPBYTE type.

pDetail: Specifies a 1-byte memory address that sets detailed information after a returned reading error in the case that addition of detailed information is specified by the BiMICRSelectDataHandlingfunction. See BiMICRSetReadBackFunction concerning the MICR detailed information that is returned. This is a LPBYTE type.

timeout: Specifies the timeout time for data reading in ms units. This is a DWORD type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer’s power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

If this function is called in the case that check reading is not being executed by the BiMICRReadCheckfunction, the reading data are not set. The reading status ends with an error.

Cautions

Depending on the printer model (firmware version), the detailed information after there is a reading error may not be supported. In that case, nothing will be set.

If the reading data overflows, bit 6 of the MICR reading status goes ON (reading data overflow).

BiMICRSelectDataHandling

Selects the check reading operation.

Syntax

nErr = **BiMICRSelectDataHandling** (*nHandle* , *charSelect* , *detailSelect* , *errorSelect*)

Argument

nHandle: Specifies the handle. This is an INT type.

charSelect: Specifies handling of characters that cannot be analyzed. This is a BYTE type.

Value	Description
0	Interrupts analysis processing at the point when characters that cannot be analyzed are detected and doesn't 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 BiMICRSetReadBackFunction / BiMICRSetReadBackWnd, the reading data are added.

detailSelect: Specifies whether or not to add detailed information after a reading error. This is a BYTE type.

Value	Description
0	Detailed information is not added.
1	Detailed information is added.

errorSelect: Specifies whether or not to end the MICR function or continue it after an error. Furthermore, the MICR function continues regardless of this setting if reading ends normally or if there was an error in adding the reading results. This is a BYTE type.

Value	Description
0	The MICR function is ended only after there is an error in not adding the reading data.
1	If reading ends due to an error caused by any of the following causes, the MICR function continues even after notification of the reading results. * A check with a non-standard length is inserted. * The magnetic waveform cannot be detected. * Characters which cannot be analyzed are detected in analysis processing. * Errors were detected in the noise measurements.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiMICRSetReadBackFunction

Executes reading of checks by BiMICRReadCheck and registers the address of the callback function when the results are notified as well as the memory addresses where each type of information read from the check is set.

> **NOTE**
> This is unavailable when the development environment is VB.

Syntax

nErr = **BiMICRSetReadBackFunction** (*nHandle , int (CALLBACK EXPORT *pMicrCB)(void) , pReadBuffSize , readCharBuff , pStatus ,pDetail)*

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pMicrCB)(void):
Specifies the address of the callback function for notifying the results from reading of a check.

pReadBuffSize: Specifies the size of the memory where the reading data are set. After execution of this function, the size of the data which were actually read is set. This is a LPBYTE type.

readCharBuff: Specifies the memory address where the check reading data are set. This is a LPBYTE type.

pStatus: Specifies a 1-byte memory address where the reading status is stored. See MICR reading status below concerning the status that is returned. This is a LPBYTE type.

For the TM-H6000, TM-H6000II, TM-H6000III, TM-J7000/7100

Bit	ON/OFF	Value	Status
0	ON	0x01	Reading font: CMC7
	OFF	0x00	Reading font: E13B
1	-	0x00	Fixed to 0
2	-	0x00	Fixed to 0
3	ON	0x08	Detailed information: Added
	OFF	0x00	Detailed information: Not added
4	ON	0x10	Rereading: Disabled
	OFF	0x00	Rereading: Enabled
5	ON	0x20	Reading result: Abnormal end
	OFF	0x00	Reading result: Normal end
6	ON	0x40	Reading data overflow: YES
	OFF	0x00	Reading data overflow: NO

For the TM-H6000, TM-H6000II, TM-H6000III, TM-J7000/7100

Bit	ON/OFF	Value	Status
7	-	0x00	Fixed to 0

For the TM-H5000II, TM-H5200, TM-U675, TM-U950

Bit	ON/OFF	Value	Status
0	ON	0x01	Reading font: CMC7
	OFF	0x00	Reading font: E13B
1	-	0x00	Fixed to 0
2	-	0x00	Fixed to 0
3	-	0x00	Fixed to 0
4	ON	0x10	Rereading: Disabled
	OFF	0x00	Rereading: Enabled
5	ON	0x20	Reading result: Abnormal end
	OFF	0x00	Reading result: Normal end
6	ON	0x40	Reading data overflow: YES
	OFF	0x00	Reading data overflow: NO
7	-	0x00	Fixed to 0

pDetail: Specifies a 1-byte memory address in which reading of a check ends in an error, which is returned in cases where detailed information is added in accordance with the BiMICRSelectDataHandlingfunction. See the following detailed MICR information concerning the detailed information that is returned. This is a LPBYTE type.

Value	Information
40h	No abnormality
41h	Check reading was not executed even once.(The BiMICRREADCheck-function has not been called.)
42h	Check insertion wait was canceled.(The BiMICRCancelWaitCheckInsertionfunction was called.)
43h	Check insertion wait was canceled by the set time being exceeded. (The time out set time passed while the BiMICRReadCheckfunction was being called.)
44h	A check with a non-standard length was inserted.
45h	The magnetic waveform was not detected.
46h	Characters which could not be analyzed were detected in analysis processing.
47h	An error occurred during check reading processing.
48h	An error was detected in the noise measurement.
49h	Check reading processing was interrupted by the cover being opened.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

Be sure to call this function before calling the BiMICRReadCheckfunction. An error will be returned if the BiMICRReadCheckfunction is called before this function is called.

Depending on the printer model (firmware version), the detailed information after there is a reading error may not be supported. In that case, nothing will be set.

If the reading data overflows, bit 6 of the MICR reading status goes ON (reading data overflow).

BiMICRSetReadBackFunctionEx

Executes reading of checks by BiMICRReadCheck and registers the address of the callback function when the results are notified as well as the memory addresses where each type of information read from the check is set.

NOTE This is unavailable when the development environment is VB.

Syntax

nErr = **BiMICRSetReadBackFunctionEx** *(nHandle , int (CALLBACK EXPORT *pMicrCB) (LPSTR lpcPortName) , pReadBuffSize , readCharBuff , pStatus ,pDetail)*

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pMicrCB)(LPSTR lpcPortName):

Specifies the address of the callback function for notifying the results from reading of a check.

pReadBuffSize: Specifies the size of the memory where the reading data are set. After execution of this function, the size of the data which were actually read is set. This is a LPBYTE type.

readCharBuff: Specifies the memory address where the check reading data are set. This is a LPBYTE type.

pStatus: Specifies a 1-byte memory address where the reading status is stored. See MICR reading status below concerning the status that is returned. This is a LPBYTE type.

pDetail: Specifies a 1-byte memory address in which reading of a check ends in an error, which is returned in cases where detailed information is added in accordance with the BiMICRSelectDataHandlingfunction. See the following detailed MICR information concerning the detailed information that is returned.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

The following MICR reading states can be acquired. Refer to [Explanation] and [Caution] in BiMICRSetReadBackFunction.

Caution

The difference between this argument and BiMICRSetReadBackFunction is that the port from which the callback is sent can be recognized.

BiMICRSetReadBackWnd

Executes check reading by BiMICRReadCheck, registers the handle of a button which sends the button's click event when the results are notified, and registers the memory addresses where each type of reading information is set.

Syntax

nErr = **BiMICRSetReadBackWnd** (*nHandle, hWnd, pReadBuffSize, readCharBuff, pStatus, pDetail*)

Argument

nHandle: Specifies the handle. This is an INT type.

hWnd: Specifies the button handle. Sends the click event to this button.
This is a Long type.

pReadBuffSize: Specifies the size of memory where the reading data are set. after this function is executed, the size of the actual reading data is set. This is a LPBYTE type.

readCharBuff: Specifies the memory addresses where the reading data are set. This is a LPBYTE type.

pStatus: Specifies a 1-byte memory address where the reading status is set. This is a LPBYTE type.

pDetail: Specifies a 1-byte memory address that sets detailed information after a returned reading error in the case that addition of detailed information is specified by the BiMICRSelectDataHandlingfunction. This is a LPBYTE type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	The specified data already exists.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Acquires the MICR reading status of 5.19 “BiMICRSetReadBackFunction.”

Cautions

Be sure to call this function before calling the BiMICRReadCheckfunction. An error will be returned if the BiMICRReadCheckfunction is called before this function is called.

Depending on the printer model (firmware version), the detailed information after there is a reading error may not be supported. In that case, nothing will be set.

If the reading data overflows, bit 6 of the MICR reading status goes ON (reading data overflow).

BiMICRCancelReadBack

Cancels a reading information notification request registered using BiMICRSetReadBackFunction, BiMICRSetReadBackFunctionEx or BiMICRSetReadBackWnd.

Syntax

nErr = **BiMICRCancelReadBack** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiSCNClumpPaper

Sets the cut sheet paper in the clumped state. After this function is executed, the printer is set in the cut sheet insertion wait state, then the paper is clumped after it is inserted. If paper is already clumped at this time, nothing is done.

Syntax

nErr = **BiSCNClumpPaper** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends normally, but the paper is not clumped.

This function is valid for the scanner unit currently selected.

BiSCNGetClumpStatus

Acquires the clump status of cut sheets.

Syntax

nErr = **BiSCNGetClumpStatus** (*nHandle* , *pStatus*)

Argument

nHandle: Specifies the handle. This is an INT type.

pStatus: Specifies the memory address where the cut sheet clump status is to be stored. See the cut sheet clump status items shown below concerning the returned status. This is a LPBYTE type.

Bit	ON/OFF	Value	Status
0	ON	0x01	Feeding processing of slip to printing position: Not possible
	OFF	0x00	Feeding processing of slip to printing position: Possible
1	ON	0x02	Reading MICR of check: Not possible
	OFF	0x00	Reading MICR on check: Possible
2	ON	0x04	Image processing of check: Not possible
	OFF	0x00	Image processing of check: Possible
3	ON	0x08	Image processing of card scanner unit: Not possible
	OFF	0x00	Image processing of card scanner unit: Possible
4	-	0x00	Fixed to 0
5	-	0x00	Fixed to 0
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported

Macro Definition (Constant)	Value	Description
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

BiSCNGetCroppingArea

Acquires the cropping area set in the printer.

Syntax

nErr = **BiSCNGetCroppingArea** (*nHandle* , *pBuffSize* , *pBuff*)

Argument

nHandle: Specifies the handle. This is an INT type.

pBuffSize: Specifies the size of memory where the cropping area information is set. The actual size of the scanned data is set. This is a LPWORD type.

pBuff: Specifies the memory address where the cropping area information is set. The information is set in the following format. This is a LPBYTE type.
The following format is returned.

Cropping area number (1 byte)
Starting X-coodinate of cropping area (1 byte)
Starting Y-coodinate of cropping area (1 byte)
Ending X-coodinate of cropping area (1 byte)
Ending Y-coodinate of cropping area (1 byte)
:
:

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_BUFFER_OVER_FLOW	-140	Lack of buffer capacity.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

Macro Definition (Constant)	Value	Description
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

If the buffer capacity is insufficient after this function is executed, the necessary number of bytes is set in pBuffSize. If any other error occurs, nothing is set.

This function is valid only for check paper.

BiSCNSetCroppingArea

Specifies the cropping area.

Syntax

nErr = **BiSCNSetCroppingArea** (*nHandle ,bAreaNo , bStartX , bStartY , bEndX , bEndY*)

Argument

nHandle: Specifies the handle. This is an INT type.

bAreaNo: Specifies the cropping area number. The valid setting range is 1~255. This is a BYTE type.

bStartX: Specifies the starting X coordinate of the cropping area. The valid setting range is 0 ~ 254 in mm units.
This is a BYTE type.

bStartY: Specifies the starting Y coordinate of the cropping area. The valid setting range is 0 ~ 254 in mm units.
This is a BYTE type.

bEndX: Specifies the ending X coordinate of the cropping area. The valid setting range is 1 ~ 255 in mm units.
This is a BYTE type.

bEndY: Specifies the ending Y coordinate of the cropping area. The valid setting range is 1 ~ 255 in mm units.
This is a BYTE type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

Macro Definition (Constant)	Value	Description
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

For the TM-H6000III, which is not equipped with a scanner, this function ends normally even if it is executed, but the range is not set.

The coordinate reference point (0, 0) is the top right when facing toward the check paper insertion direction.

If bStartX > = bEndX, or bStartY > = bEndY, it results in a parameter error.

Valid ranges for arguments of BiSCNSetCroppingArea are as follows. A value exceeding the range cannot be set.
1 <= bAreaNo <= 10, 0 <= bStartX <= 98, 0 <= bStartY <= 228, 2 <= bEndX <= 255, 2 <=bEndY <= 255

If the specified Cropping area number already exists, the cropping area with that number is updated.

If a bEndX, bEndY which exceeds the printer’s scannable area is specified, the values for the printer’s maximum scannable area are set.

If an odd number is specified to a set value, it is handled as an even number. A starting point (bStarX, bStartY) is handled on the minus side, and an ending point (bEndX, bEndY) is handled on the plus side.

This function is valid only for check paper.

The set read area is applied beginning with the next image data read. (The read area is not applied to image data that has already been read

BiSCNDeleteCroppingArea

Deletes the cropping area set in the printer.

Syntax

nErr = **BiSCNDeleteCroppingArea** (*nHandle* ,*bAreaNo*)

Argument

nHandle: Specifies the handle. This is an INT type.

bAreaNo: Specifies the number of the cropping area to be deleted. The valid setting range is 1-255. If 0 is set, all the cropping areas are deleted. This is a BYTE type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends normally, but the cropping area is not deleted.

This function is valid only for check paper.

BiSCNGetImageFormat

Acquires the format of the image set in the printer.

Syntax

nErr = **BiSCNGetImageFormat** (*nHandle* , *pFormat*)

Argument

nHandle: Specifies the handle. This is an INT type.

pFormat: Specifies the memory address where the format of the notified image data is set. The set value is as shown below. This is a LPBYTE type.

Macro Definition (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

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiSCNSetImageFormat

Sets the format of the scanning image data. Furthermore, the value set with this function is valid until the printer's power is turned OFF or until the printer is reset.

Syntax

nErr = **BiSCNSetImageFormat** (*nHandle* , *bFormat*)

Argument

nHandle: Specifies the handle. This is an INT type.

bFormat: Specifies the format of image data notified. The valid specification values are as shown below. This is a BYTE type.
The default value is EPS_BI_SCN_TIFF after the printer's power is turned on, or after it is reset.

Macro Definition (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

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends normally, but the format is not set.

Since this function does not check whether the scanner of the printer is capable of using a specified image data format, be sure to confirm it with the information acquired by the function of BiGetPrnCapability before using.

This function is valid for the scanner unit currently selected.

The set reading quality is applied beginning with the next image data read. (Reading quality is not applied to image data that has already been read.)

BiSCNGetImageList

Acquires a data ID list of the reading result of the images saved in the NV memory. If there is no data ID saved, it returns 0 to pListNum.

Syntax

nErr = **BiSCNGetImageList** (*nHandle , pListNum , pIDList*)

Argument

nHandle: Specifies the handle. This is an INT type.

pListNum: Specifies the number of data ID acquired in the data ID list. After executing this function, the actual total number of the data ID acquired is set. If the number of the data ID saved is bigger than the number of data ID specified, the number of data ID required is set and in the case of the error other than this, nothing is set. This is a LPWORD type.

pIDList: Specifies the memory address for setting the data ID list. Information should be set with the format below.

DataID1 (1-65535) , ... , Data IDk
This is a LPWORD type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer’s power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_BUFFER_OVER_FLOW	-140	Lack of buffer capacity.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting

Macro Definition (Constant)	Value	Description
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

Data ID saved in the work area cannot be acquired.

BiSCNGetImageQuality

Acquires the scanning quality of the image set in the printer.

Syntax

nErr = **BiSCNGetImageQuality** (*nHandle* , *pColorDepth* , *pThreshold* , *pColor* , *pExOption*)

Argument

nHandle: Specifies the handle. This is an INT type.

pColorDepth: Specifies the memory address where the tonal gradation is set. This is a LPBYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_1BIT	1	1bit
EPS_BI_SCN_8BIT	8	8bit

pThreshold: Specifies the memory address where the density threshold value is set. This is a CHAR type.

pColor: Specifies the memory address where color is set. The set value is as shown below. This is a LPBYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_MONOCHRO3E	48	Black and White
EPS_BI_SCN_COLOR	49	Color

pExOption: Specifies a memory address to which a density adjustment type is set. Set values are as shown below. This is a LPBYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_MANUAL	49	Density is adjusted manually.
EPS_BI_SCN_SHARP	50	Sharpening

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for a TM-H6000III that is not equipped with a scanner, it ends with an ERR_TIMEOUT.

This function is valid for the scanner unit currently selected.

BiSCNSetImageQuality

Sets the image scanning quality. Furthermore, the value set with this function is valid until the printer’s power is turned OFF or until the printer is reset.

Syntax

nErr = **BiSCNSetImageQuality** (*nHandle* , *bColorDepth* , *bThreshold* , *bColor* , *bExOption*)

Argument

nHandle: Specifies the handle. This is an INT type.

bColorDepth: Specifies the tonal gradation (the number of bits used for 1 pixel). The valid specification values are 1 or 8. The default value after the printer’s power is turned off, or after the printer is reset, is 1. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_1BIT	1	1bit
EPS_BI_SCN_8BIT	8	8bit

bThreshold: Specifies the density threshold value. The valid specification range is -128 - 127. The default value after the printer’s power is turned off, or after the printer is reset, is 0. If 0 is specified, the printer is set to use the basic density. If -128 is specified, the printer calculates the optimum value automatically and use it as the density threshold value. This is a CHAR type.

bColor: Specifies the memory address where color is set. The set value is as shown below. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_MONOCHROME	48	Monochrome
EPS_BI_SCN_COLOR	49	Color

bExOption: Specifies density adjustment types. Values that can be specified are as shown below. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_MANUAL	49	Density is adjusted manually.
EPS_BI_SCN_SHARP	50	Sharpening

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Coution

If this function is executed for a TM-H6000III that is not equipped with a scanner, it ends normally, but the image quality is not set.
Since this function does not check whether the scanner of the printer is capable of gray scale reading, be sure to confirm this before using a printer with the information acquired by using the function BiGetPrnCapability.
Only when pColorDepth is EPS_BI_SCN_1BIT(1), the value of *pThreshold is ignored.
This function is valid for the scanner unit currently selected.

The set reading quality is applied beginning with the next image data read. (Reading quality is not applied to image data that has already been read.)

BiSCNGetImageRemainingCapacity

Acquires the remaining capacity (in bytes) of the NV memory for saving the reading result of images.

Syntax

nErr = **BiSCNGetImageRemainingCapacity** (*nHandle* , *pSize*)

Argument

nHandle: Specifies the handle. This is an INT type.

pSize: Specifies the memory address for setting the remaining capacity of the NV memory for saving the reading result of images. This is a LPDWORD type.

Return Values

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

The remaining capacity is the remaining when subtracting fail block, memory management area, and the reading result of images from all the memory capacity for saving the reading result of images.

BiSCNGetScanArea

Acquires the image scanning area specified in the printer.

Syntax

nErr = **BiSCNGetScanArea** (*nHandle* , *pStartX* , *pStartY* , *pEndX* , *pEndY*)

Argument

nHandle: Specifies the handle. This is an INT type.

pStartX: Specifies the memory address where the starting X coordinate of the scanning area is set. This is a LPBYTE type.

pStartY: Specifies the memory address where the starting Y coordinate of the scanning area is set. This is a LPBYTE type.

pEndX: Specifies the memory address where the ending X coordinate of the scanning area is set. This is a LPBYTE type.

pEndY: Specifies the memory address where the ending Y coordinate of the scanning area is set. This is a LPBYTE type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer’s power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Comment

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

This function is valid only for scanning check paper.

BiSCNSetScanArea

Sets the image scanning area. Furthermore, the values set with this function remain in effect until the printer's power is turned off or until the printer is reset.

Syntax

nErr = **BiSCNSetScanArea** (*nHandle* , *bStartX* , *bStartY* , *bEndX* , *bEndY*)

Argument

nHandle: Specifies the handle. This is an INT type.

bStartX: Specifies the starting X coordinate of the scanning area. The valid specification range is 0-254 in mm units.(The valid specification range in the TM-H6000III is 0-99.)
The default value after the printer's power is turned off or after printer reset is 0.
This is a BYTE type.

bStartY: Specifies the starting Y coordinate of the scanning area.
The valid specification range is 0~254 in mm units. The default value after the printer's power is turned off or after printer reset is 0.
This is a BYTE type.

bEndX: Specifies the ending X coordinate of the scanning area. The valid specification range is 0-255 in mm units.(The valid specification range in the TM-H6000III is 0-100.)
If 0 is specified, that model's maximum value is used.
The default value after the printer's power is turned off or after printer reset is 70.
This is a BYTE type.

bEndY: Specifies the ending Y coordinate of the scanning area. The valid specification range is 0-255 in mm units.
If 0 is specified, that model's maximum value is used.
The default value after the printer's power is turned off or after printer reset is 0.
This is a BYTE type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends normally, but the scan area is not set.

The coordinate reference point (0, 0) is the top right when facing toward the check paper insertion direction.

If bStartX > = bEndX, or bStartY > = bEndY (bEndY is not 0), it results in a parameter error.

If a bEndX, bEndY which exceeds the printer's scannable area is specified, the values for the printer's maximum scannable area are set.

The maximum value that can be set may differ depending on the printer model with built-in scanner. Therefore, in the Status API, a check of the scanning area values is not conducted for every model. As a result, the specified value is transmitted as is to the printer and operation is according to the command interpretation specifications of each printer.

Example: If the end X coordinate 101 is specified for the scanning area of the H6000III, (The original valid specification range is 0 ~ 100.)

1)The API transmits 101 as the scanning area's bEndX coordinate to the printer.

2)The printer regards the value as having exceeded the scannable area and sets the value to the maximum value 100.

This function is valid only for scanning check paper.

The set read area is applied beginnig with the next image data read. (The read area is not applied to image data that has already been read.)

When an odd number is specified for the setting value, it will be rounded to an even number. The starting point bStartX, bStartY will be rounded to the minus side (rounded downwards), and the end point bEndX, bEndY will be rounded to the plus side (rounded upwards).

BiSCNPreScan

This sets the printer's density threshold value after prescanning of the image and returns the set density threshold value. (An image can be scanned at the threshold value obtained from the prescan operation only in the case that this function is called and the image is scanned without ejecting the paper (BiSCNSetImageQuality).)

Syntax

nErr = **BiSCNPreScan** (*nHandle* , *pResultStatus* , *pThreshold*)

Argument

nHandle: Specifies the handle. This is an INT type.

pResultStatus: Specifies the memory address where the prescan results status is set. The set value is as shown below. This is a LPBYTE type.

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	-	0x00	Fixed to 0
5	ON	0x20	Ends with an error
	OFF	0x00	Ends normally
6	-	0x40	Fixed to 1
7	-	0x00	Fixed to 0

pThreshold: This specifies the memory address where the prescanning results, the threshold value of the density of the paper's automatically measured density, is set. The value is set if the prescanning results are successful, and if the prescanning results are not successful, no value is set, so this value becomes indefinite. It is necessary to judge the Prescan results from PresultStatus. This is a CHAR type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer’s power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

BiSCNReadImage

Executes image scanning. If 0 is specified for the paper insertion wait time, the printer wait time is canceled.

Syntax

nErr = **BiSCNReadImage** (*nHandle* , *wId* , *bSelectSheet* , *bWaitInsertionTime* ,
bAddInforDataSize , *pAddInforData* ,*bMemory*)

Argument

nHandle: Specifies the handle. This is an INT type.

wId: Specifies the data identifier related to the scanned image. Data identifier can be specified in the range of 1- 65535.
This is a WORD type.

bSelectSheet: Specifies the paper which is the scanning object. The types that can be specified are as shown below.
However, in the current version, this setting is fixed on EPS_BI_SCN_CHECKPAPER, and any other value that is specified is regarded as invalid. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_JOURNAL	1	Journal
EPS_BI_SCN_RECEIPT	2	Receipt
EPS_BI_SCN_CARD	16	Card
EPS_BI_SCN_CHECKPAPER	32	Check paper

bWaitInsertionTime: Specifies the paper insertion wait time. 0 ~ 15 [x 60 sec.] (The default is 0.) If 0 is specified, the wait time is 30 seconds. This is a BYTE type.

bAddInforDataSize: Specifies the size of character data if character information is added to an image. This is a BYTE type.

pAddInforData: Specifies the memory address where character data are set if character information is added to an image. This is a LPBYTE type.

bMemory: Specifies whether to save or not save the result of reading images in the NV memory. This is a BYTE type.

Macro Definition (Constant)	Value	Description
EPS_BI_SCN_NVMEMORY _NOTSAVE	48	not saved (reading result is saved in the work area)
EPS_BI_SCN_NVMEMORY_SAVE	49	saved

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_WITHOUT_CB	-130	Cannot execute because either BiSCNSetReadBackFunction or BiSCNSetReadBackWnd has not been called.
ERR_PAPERINSERT_TIMEOUT	-300	Paper insertion waiting time has passed.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends normally, but image scanning is not executed.

Before using this function, be sure to register it using either BiSCNSetReadBackFunction or BiSCNSetReadBackWnd.

After image scanning is completed, the check paper insertion read time is reset to the default value 0.

If the data identifier that is already present is specified, the saved image is deleted and replaced with a new image.

Since this function does not check whether NV memory in the printer for saving the reading result of an image is present or not, be sure to confirm it with the information acquired by the function of BiGetPrnCapability before using.

The setting for paper insertion waiting time using bWaitInsertionTime cannot currently be used.

The other functions cannot be used until the paper is inserted after BiSCNReadImage is executed.

If the following formats are selected, characters cannot be added to data.

- Noncompressed data in raster format
- Noncompressed data in bit map format

A value of a parameter that can be specified to paper used for scanning differs, depending on the unit currently selected. Effective combinations are listed below. Combinations other than those described below are parameter errors. Refer to BiSelectScanUnit for a setting method of a selected unit.

bSelectSheet (Macro Definition)	Selected unit
EPS_BI_SCN_CHECKPAPER	When a selected unit is a unit for check reading
EPS_BI_SCN_CARD	When a selected unit is a unit for card scanning

BiSCNRetransmissionImage

Retransmits the image scanning results.

Syntax

nErr = **BiSCNRetransmissionImage** *(nHandle, wID , pBuffSize, pBuff, pImageXsize, pStatus,pDetail ,dwTimeout)*

Argument

nHandle: Specifies the handle. This is an INT type.

wId: Specifies the data identifier related to the scanned image. Data identifier can be specified in the range of 1- 65535.
This is a WORD type.

pBuffSize: Specifies the size of memory where the image data are set. After this function is executed, the size of the actually scanned data is set.
This is a LPDWORD type.

pBuff: Specifies the memory address where the image data are set. This is a LPBYTE type.

pImageXsize: Specifies the memory address where the number of image data (bytes) in the X direction is set.
After reading, if the number of horizontal data is 128 or more, the actual number of horizontal data is produced by subtracting “128” from the number of horizontal data and multiplying it by 8. If the number of data does not exceed 128, the number set is the actual number of horizontal data. (For example, the number of horizontal data acquired is 198, the actual number of horizontal data is 560 by following the calculation: (198-128)*8 = 560. This is a LPBYTE type.

pStatus: Specifies the memory address where the read status is stored. Concerning the status that is returned, see the BiSCNSetReadBackFunction image scanning status. This is a LPBYTE type.

pDetail: Specifies the memory address where detailed information is set when scanning ends with an error. Concerning the status that is returned, see the BiSCNSetReadBackFunction image scanning status. This is a LPBYTE type.

dwTimeout: Specifies the data reading time out time in ms units. This time out is a value that is measured from the point when there is no response from the printer after a rereading request. This is a DWORD type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_WITHOUT_CB	-130	Cannot execute because either BiSCNSetReadBackFunction or BiSCNSetReadBackWnd has not been called.
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends with an ERR_TIMEOUT.

If this function is called by the BiSCNReadImage function when no image scanning is being executed or a specified data identifier is not present, the scanning data are not set (the read status shows that it ended in an error).

If the scanning data overflows, the 6th bit of the read status goes ON (read data overflow). In that case, the "required number of bytes" is set in pBuffSize.

Sufficient buffer capacity should be specified in pBuffSize and pBuff. The maximum image data size is 4 megabytes.

BiSCNDeleteImage

Deletes the reading result of an image of a specified data ID or deletes the reading result of all data IDs saved at a time.
When executing a collective deletion, both the NV memory used for image saving and the reading result of images saved in the work area are deleted.

Syntax

nErr = **BiSCNDeleteImage** (*nHandle ,wID*)

Argument

nHandle: Specifies the handle. This is an INT type.

wID: Specifies data ID to be deleted. A value can be specified in the range of 1 - 65535. If 0 is specified to the data ID, all the reading result of the ID images saved is deleted at the same time. This is a WORD type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer’s power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

If this function is executed for the TM-H6000III, which is not equipped with a scanner, it ends normally, but the scanning data is not deleted.

This function is valid only for check paper.

BiSCNSelectScanUnit

Sets the unit that operates image scanning.

Syntax

nErr = **BiSCNSelectScanUnit** (*nHandle, bSelectUnit*)

Argument

nHandle: Specifies the handle. This is an INT type.

bSelectUnit: Specifies the unit to scan. This is a BYTE type. The selectable value is as follows.

Macro Definition (Constant)	Value	Unit
EPS_BI_SCN_CHECKPAPER	48	Check paper reading unit
EPS_BI_SCN_CARD	49	Card reading unit

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	A time out error occurred.
ERR_ACCESS	-80	Reading/writing with the printer is not possible (The printer's power is not on or the cable connections are faulty or printing or accessing, etc.).
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_MICR	-320	Cannot call as the MICR is reading
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning
ERR_PH_NOT_EXIST	-340	The PortHandler is not running or a communication error between the client of PortHandler and the server.
ERR_RESET	-400	Cannot call as the printer is restarting
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Sets EPS_BI_SCN_UNIT_CHECKPAPER for the initial value. The functions below are effective to the unit currently selected.

- BiSCNSetImageFormat
- BiSCNGetScanArea
- BiSCNSetScanArea
- BiSCNGetImageQuality
- BiSCNSetImageQuality
- BiSCNGetCroppingArea
- BiSCNSetCroppingArea
- BiSCNDeleteImage

BiSCNSetReadBackFunction

Executes image scanning by the function BiSCNReadImage and registers the callback function's address called when sending notification of the results as well as the memory addresses for setting each type of scanned information.

Syntax

nErr = **BiSCNSetReadBackFunction** (*nHandle , int (CALLBACK EXPORT *pScnCB)(void) , pBuffSize , pBuff , pImageXsize , pStatus ,pDetail)*

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pScnCB)(void):
Specifies the callback function address for sending notification of the results of image scanning.

pBuffSize: Specifies the size of memory where image data are to be set. After execution of this function, the actual size of the scanned data is set. This is a LPDWORD type.

pBuff: Specifies the memory address where image data are set. This is a LPBYTE type.

pImageXsize: Specifies the memory address where the number of data (bytes) of image data in the X direction is set.
If the number of horizontal data is 128 or more, the actual number of horizontal data is produced by subtracting "128" from the number of horizontal data and multiplying it by 8. If the number of data does not exceed 128, the number set is the actual number of horizontal data. (For example, the number of horizontal data acquired is 198, the actual number of horizontal data is 560 by following the calculation: (198-128)*8 = 560.
This is a LPBYTE type.

pStatus: Specifies the memory address where the read status is stored. Concerning the returned status, refer to the image scanning status items below. This is a LPBYTE type.

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	ON	0x10	Rereading: Disabled (Fixed)
	OFF	0x00	Rereading: Enabled
5	ON	0x20	Reading result: Abnormal end
	OFF	0x00	Reading result: Normal end
6	ON	0x40	Reading data overflow: YES
	OFF	0x00	Reading data overflow: NO
7	ON	0x80	Reading data reception error: YES
	OFF	0x00	Reading data reception error: NO

pDetail: Specifies the memory address where detailed information is set after scanning ends with an error. Concerning the returned status, see the detailed image information below. This is a LPBYTE type.

Value	Description
40h	No error.
41h	The image scanning result do not exist
44h	The cover was opened, so image scanning was interrupted
45h	A recoverable error/automatic reset error occurred during image scanning.
46h	Paper which is not the standard length was inserted (longer than approximately 333 mm).
47h	Compressed data error (The amount of data increased in data compression processing, and there was insufficient memory.)
48h	Paper insertion status or paper feed error
60h	Lack of remaining capacity in NV memory for saving reading result of images
61h	Failure of writing process of reading result of images to the NV memory
62h	Failure of deletion process of the NV memory for reading result of images

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_NO_MEMORY	-50	Memory is insufficient.
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	Data selected is already present
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

This function should always be called before the BiSCNReadImage function is called. If the BiSCNReadImage function is called before this function, it will result in an error.

If the scanning data overflow, the 6th bit of the read status goes ON (read data overflow). In that case, the “required number of bytes” is set in pBuffSize.

Sufficient buffer capacity should be specified in pBuffSize and pBuff. The maximum image data size is 4 megabytes.

BiSCNSetReadBackFunctionEx

Executes image scanning by the function BiSCNReadImage and registers the callback function's address called when sending notification of the results as well as the memory addresses for setting each type of scanned information.

> NOTE
>
> This is unavailable when the development environment is VB.

Syntax

nErr = **BiSCNSetReadBackFunctionEx** (*nHandle , int (CALLBACK EXPORT *pScnCB)*
(LPSTR lpcPortName) , pBuffSize , pBuff ,
pImageXsize , pStatus ,pDetail)

Argument

nHandle: Specifies the handle. This is an INT type.

int (CALLBACK EXPORT *pScnCB)(LPSTR lpcPortName):
Specifies the callback function address for sending notification of the results of image scanning. The port name is set to lpcPortName and a callback argument is called.

pBuffSize: Specifies the size of memory where image data are to be set. After execution of this function, the actual size of the scanned data is set. This is a LPDWORD type.

pBuff: Specifies the memory address where image data are set. This is a LPBYTE type.

pImageXsize: Specifies the memory address where the number of data (bytes) of image data in the X direction is set.
If the number of horizontal data is 128 or more, the actual number of horizontal data is produced by subtracting "128" from the number of horizontal data and multiplying it by 8. If the number of data does not exceed 128, the number set is the actual number of horizontal data. (For example, the number of horizontal data acquired is 198, the actual number of horizontal data is 560 by following the calculation: (198-128)*8 = 560. This is a LPBYTE type.

pStatus: Specifies the memory address where the read status is stored. Concerning the returned status, refer to the image scanning status items below. This is a LPBYTE type.

pDetail: Specifies the memory address where detailed information is set after scanning ends with an error. Concerning the returned status, see the detailed image information below. This is a LPBYTE type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_NO_MEMORY	-50	Memory is insufficient.
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	Data selected is already present
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

This function should always be called before the BiSCNReadImage function is called. If the BiSCNReadImage function is called before this function, it will result in an error.

If the scanning data overflow, the 6th bit of the read status goes ON (read data overflow). In that case, the "required number of bytes” is set in pBuffSize.

Sufficient buffer capacity should be specified in pBuffSize and pBuff. The maximum image data size is 4 megabytes.

The difference between this argument and BiSCNSetReadBackFunction is that the port from which the callback is sent can be recognized.

Refer to BiSCNSetReadBackFunction for Scanner reading status.

BiSCNSetReadBackWnd

This function executes image scanning with the BiSCNReadImage function and registers the memory address for the handle of the button where button click events are sent when sending notification of the scanning results as well the memory addresses for setting each type of information.

Syntax

nErr = **BiSCNSetReadBackWnd** (*nHandle, hWnd, pBuffSize, pBuff, pImageXsize, pStatus,pDetail)*

Argument

nHandle: Specifies the handle. This is an INT type.

hWnd: The button handle. Click events are sent to this button.
This is a Long type.

pBuffSize: Specifies the size of memory where image data are sent. The actual scanning data size is set after this function is executed. This is a LPDWORD type.

pBuff: Specifies the memory address where image data are set. This is a LPBYTE type.

pImageXsize: Specifies the memory address where the number of data (bytes) in the X direction of the image data is set.
After reading, if the number of horizontal data is 128 or more, the actual number of horizontal data is produced by subtracting "128" from the number of horizontal data and multiplying it by 8. If the number of data does not exceed 128, the number set is the actual number of horizontal data. (For example, the number of horizontal data acquired is 198, the actual number of horizontal data is 560 by following the calculation: (198-128)*8 = 560. This is a LPBYTE type.

pStatus: Specifies the memory address where the scanning status is stored. See the BiSCNSetReadBackFunction image scanning status concerning the returned status. This is a LPBYTE type.

pDetail: Specifies the memory address where the detailed information is set when scanning ends with an error. See the BiSCNSetReadBackFunction image detailed information concerning the returned status. This is a LPBYTE type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_NO_MEMORY	-50	Memory is insufficient.
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXIST	-210	Data selected is already present
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Caution

This function should always be called before the BiSCNReadImage function is called. If the BiSCNReadImage function is called before this function, it will result in an error.

If the scanning data overflow, the 6th bit of the read status goes ON (read data overflow). In that case, the “required number of bytes” is set in pBuffSize.

Sufficient buffer capacity should be specified in pBuffSize and pBuff. The maximum image data size is 4 megabytes.

BiSCNCancelReadBack

Cancels the scanning information notice request registered using BiSCNSetReadBackFunction, BiSCNSetReadBackFunctionEx or BiSCNSetReadBackWnd.

Syntax

nErr = **BiSCNCancelReadBack** (*nHandle*)

Argument

nHandle: Specifies the handle. This is an INT type.

Return value

If execution is successful, 0 is returned. If an error occurs, the following error codes (negative values) are returned.

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value specified for the printer is incorrect.
ERR_TIMEOUT	-70	Timeout error
ERR_ACCESS	-80	R/W cannot be performed on the printer
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_LOCKED	-1000	The printer is locked.

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiESCNEnable

Set so that scanner extended functions can be used.

Syntax

nErr = **BiESCNEnable** (*bStoreType*)

Argument

bStoreType: Select a storing method for a cropped image stored using BiESCNStoreImage. This is a BYTE type.

Macro Definition (Constant)	Value	Description
CROP_STORE_MEMORY	0	Save in memory
CROP_STORE_FILE	1	Save in a file

Return value

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

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

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

- Before calling BiOpenMonPrinter, it is necessary to enable scanner extended functions by calling this argument.
- Secure save table area with BiOpenMonPrinter.
- After calling back image data reading, process the image data acquired by the printer and save it in the WORK AREA.
- Arguments of the scanner extended functions (BiESCN-) can be used.

Note

If this argument is called after calling BiOpenMonPrinter, the scanner extended functions cannot be used and the way of saving a cropped image cannot be changed.

BiESCNGetAutoSize

Acquire the value of capAutoSize.

Syntax

nErr = **BiESCNGetAutoSize** (*nHandle , pCapAutoSize*)

Argument

nHandle: Specifies the handle. This is an INT type.

pCapAutoSize: Select a memory address to set a capAutoSize value. This is a LPBYTE type.

Macro Definition (Constant)	Value	Description
CROP_AUTOSIZE_ENABLE	1	AutoSize processing enabled
CROP_AUTOSIZE_DISABLE	0	AutoSize processing disabled

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

If "1" is selected for capAutoSize, after reading image data, AutoSize processing (cut black part of the image data off) is executed, and the processed image data is saved in the WORK AREA. The width and height of the image data are automatically set to documentWidth and documentHeight.

If "0" is selected for capAutoSize, AutoSize processing and automatic setting for the width and height of the image data are not executed.

BiESCNSetAutoSize

Select the value of capAutoSize.

Syntax

nErr = **BiESCNSetAutoSize** (*nHandle* , *bCapAutoSize*)

Argument

nHandle: Specifies the handle. This is an INT type.

bCapAutoSize: Select a value for a capAutoSize. This is a BYTE type.

Macro Definition (Constant)	Value	Description
CROP_AUTOSIZE_ENABLE	1	AutoSize processing enabled
CROP_AUTOSIZE_DISABLE	0	AutoSize processing disabled

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Acquire a value of capAutoSize (AutoSize processing flag).

If an argument other than CROP_AUTOSIZE_ENABLE or CROP_AUTOSIZE_DISABLE is selected, the error (ERR_PARAM) is returned.

The AutoSize processing flag that has been set is used in the next image data reading process (AutoSize processing is not used for the image data that have been already read and saved in the WORK AREA.)

BiESCNGetCutSize

Acquire the value of cutSize.

Syntax

nErr = **BiESCNGetCutSize** (*nHandle , pCutSize*)

Argument

nHandle:	Specifies the handle. This is an INT type.
pCutSize:	Select a memory address to set a cutSize value. This is a LPWORD type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

Explanation

Acquire the value of cutSize (size for cutting the left and right sides of an image data).

If a number other than "0" is selected for cutSize, after reading the image data, CutSize processing (cut the left and right sides of the image data for a selected size) is executed, and the processed image data is saved in the WORK AREA.

If "0" is selected for cutSize, cutSize is not executed.

BiESCNSetCutSize

Select a value of cutSize.

Syntax

nErr = **BiESCNSetCutSize** (*nHandle* , *wCutSize*)

Argument

nHandle: Specifies the handle. This is an INT type.

wCutSize: Select size of cutting the left and right sides of the image data (unit: 0.1mm). Selectable range is 0 to 1500. Numbers except “0” : CutSize processing is executed with the selected size. ”0” : CutSize processing is not executed. This is a WORD type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Set the value selected by wCutSize to cutSize.

BiESCNGetRotate

Acquire the value of capRotate.

Syntax

nErr = **BiESCNGetRotate** (*nHandle* , *pCapRotate*)

Argument

nHandle: Specifies the handle. This is an INT type.

pCapRotate: Select a memory address to set a capRotate value. This is a LPBYTE type.

Macro Definition (Constant)	Value	Description
CROP_ROTATE_ENABLE	1	Rotate processing enabled
CROP_ROTATE_DISABLE	0	Rotate processing disabled

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Acquire the value of capRotate.

If CROP_ROTATE_ENABLE is set to capRotate, after reading image data, Rotate processing (rotates the image data 90° clockwise) is executed, the processed image data is saved in the WORK AREA.

If CROP_ROTATE_DISABLE is set to capRotate, Rotate is not executed.

BiESCNSetRotate

Select a value of capRotate.

Syntax

nErr = **BiESCNSetRotate** (*nHandle* , *bCapRotate*)

Argument

nHandle: Specifies the handle. This is an INT type.

bCapRotate: Select a value for capRotate. This is a BYTE type.

Macro Definition (Constant)	Value	Description
CROP_ROTATE_ENABLE	1	Rotate processing enabled
CROP_ROTATE_DISABLE	0	Rotate processing disabled

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Set the value of capRotate.

Note

If the argument other than CROP_ROTATE_ENABLE or CROP_ROTATE_DISABLE is selected, the error (ERR_PARAM) is returned.

The Rotate processing flag that has been set is used in the next image data reading process (Rotate processing is not used for the image data that has been already read and saved in the WORK AREA.)

BiESCNGetDocumentSize

Acquire the values of documentWidth and documentHeight.

Syntax

nErr = **BiESCNGetDocumentSize** (*nHandle , pDocumentWidth , pDocumentHeight*)

Argument

nHandle: Specifies the handle. This is an INT type.

pDocumentWidth: Select a memory address to set a value of the width of the image data (unit: 0.1 mm).
This is a LPWORD type.

pDocumentHeight: Select a memory address to set a value of the height of the image data (unit: 0.1 mm).
This is a LPWORD type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Acquire the values of documentWidth and documentHeight (the width and height of the image data saved in the WORK AREA) by using the unit of 0.1 mm.

Note

The unit of the argument is 0.1 mm.

If automatic update by reading the image data or a change with BiESCNSetDocumentSize is not executed, the default value (width=680, height=1520) is acquired.

BiESCNSetDocumentSize

Select values of documentWidth and documentHeight.

Syntax

nErr = **BiESCNSetDocumentSize** (*nHandle* , *wDocumentWidth* , *wDocumentHeight*)

Argument

nHandle: Specifies the handle. This is an INT type.

wDocumentWidth: Select the width of the image data (unit: 0.1 mm).
Selectable range is 100 to 3000. This is a WORD type.

wDocumentHeight: Select the height of the image data (unit: 0.1 mm).
Selectable range is 100 to 3000. This is a WORD type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	Cannot call as the scanner is scanning

NOTE For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Set the width and height selected to documentWidth and documentHeight.

Note

The argument unit is 0.1mm.

BiESCNDefineCropArea

Register CropArea or delete all the CropArea registered.

Syntax

nErr = **BiESCNDefineCropArea** (*nHandle , bCropAreaID , wStartX , wStartY , wEndX , wEndY*)

Argument

nHandle: Specifies the handle. This is an INT type.

bCropAreaID: Select CropAreaID to be registered. Selectable values are 0 to 255.
If 2 to 225 is specified, the specified coordinate is set.
This is a BYTE type.

Macro Definition (Constant)	Value	Description
CROP_AREA_RESET_ALL	0	All the CropAreas registered are deleted.
CROP_AREA_ENTIRE_IMAGE	1	"0" is set to starting points (X,Y) of CropArea. The values of documentWidth and documentHeight are set to ending points (X,Y) respectively.

wStartX: Select a starting point (X) of CropArea. Selectable values are 0 to documentWidth-1 (unit: 0.1 mm) This is a WORD type.

wStartY: Select a starting point (Y) of CropArea. Selectable values are 0 to documentHeight-1 (unit: 0.1 mm) This is a WORD type.

wEndX: Select an ending point (X) of CropArea. Selectable values are 1 to documentWidth (unit: 0.1 mm) This is a WORD type.

Macro Definition (Constant)	Value	Description
CROP_AREA_RIGHT	65535	The value of documentWidth is set to the ending X coordinate.

wEndY: Select an ending point (Y) of CropArea. Selectable values are 1 to documentHeight (unit: 0.1 mm) This is a WORD type.

Macro Definition (Constant)	Value	Description
CROP_AREA_BOTTOM	65535	The value of documentHeight is set to the ending Y coordinate.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	ERR_EXEC_SCAN Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Select each coordinate for CropArea and register it to the CropArea definition table of status API by using CropArea ID selected.

If the data of CropAreaID selected has already been defined, the registration is executed by overwriting.

If CROP_AREA_RESET_ALL is selected for CropAreaID, all the data defined in the CropArea definition table are deleted.

Note

The number of CropArea data that can be defined is up to 255 (limited item).

The argument unit is 0.1 mm.

If wStartX >= wEndX, wStartY >= wEndY, the error (ERR_PARAM) is returned.

When BiCloseMonPrinter is being executed, all the data defined in the CropArea definition table are deleted.

BiESCNGetMaxCropAreas

Acquire a maximum number of data of CropArea that can be registered .

Syntax

nErr = **BiESCNGetMaxCropAreas** (*nHandle , pMaxCropAreas*)

Argument

nHandle: Specifies the handle. This is an INT type.

pMaxCropAreas: Select a memory address to set a maximum number of data of CropArea that can be registered. This is a LPBYTE type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	ERR_EXEC_SCAN Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

BiESCNStoreImage

Register a Crop image.

Syntax

nErr = **BiESCNStoreImage** (*nHandle , dwFileIndex , pFileID , pImageTagData , bCropAreaID*)

Argument

nHandle: Specifies the handle. This is an INT type.

dwFileIndex: Select FileIndex that is identification data of the Crop image to be saved. NULL can be selected. This is a DWORD type.

pFileID: Select FileID that is identification data of the Crop image to be saved. Selectable length of a character string is up to 64 bytes. NULL can be selected. The symbols \ / : , ; * ? " < > and | cannot be used. This is a LPSTR type.

pImageTagData: Select ImageTagData that is identification data of the Crop image to be saved. Selectable length of a character string is up to 64 bytes. NULL can be used. The symbols \ / : , ; * ? " < > and | cannot be used. This is a LPSTR type.

bCropAreaID: Select CropAreaID that has already been defined by using BiESCNDefineCropArea. Selectable values are 1 to 255. This is a BYTE type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_NO_MEMORY	-50	Lack of memory
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_DISK_FULL	-170	Lack of remaining memory
ERR_NO_IMAGE	-180	Image data is not present
ERR_ENTRY_OVER	-190	Number exceeding the maximum number of being able to be registered cannot be registered
ERR_CROPAREAID	-200	Crop area selected is not present
ERR_EXIST	-210	Data selected is already present

Macro Definition (Constant)	Value	Description
ERR_IMAGE_FILEOPEN	-230	Failure in opening image data file
ERR_IMAGE_UNKNOWNFORMAT	-240	Incorrect in formatting image data file
ERR_IMAGE_FAILED	-250	Failure in creating image data file.One of the following errors is returned
ERR_WORKAREA_NO_MEMORY	-260	Failure in creating the WORK AREA because of lack of memory.
ERR_WORKAREA _UNKNOWNFORMAT	-270	Failure in creating the WORK AREA because image when an error occurs because image data is not present in the WORKAREA or an error occurs during creation of the WORK AREA
ERR_WORKAREA_FAILED	-280	Failure in creating the WORK AREA
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	ERR_EXEC_SCAN Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Crop the CropArea from the image data saved in the WORK AREA by using CropAreaID selected and register the data to the Crop image saving table.

The cropped data are the same format as the original image data; however, if they are JPEG format, they are compressed and saved with JPEG standard compression.

If CropArea exceeds the range of image data of the WORK AREA, the excess area is solid white.

The cropped image data are saved in memory or a file by using the saving method selected with BiESCNEnable.

When saving image data in a file, a file name is created by a printer handle and identification data and is saved in the Windows temporary folder.

Example: Printer handle(nHandle)=1, fileIndex=1, fileID=”AA”, imageTagData=”BBB”,

When Image format=JPEG:

File name: C:\Windows\Temp\1ESCN1_AA_BBB.jpg

When imageTagData=NULL, the file name is as follows:

File name: C:\Windows\Temp\1ESCN1_AA_.jpg

Note

Crop image data that can be saved is up to 255 (limited item). The error (ERR_ENTRY_OVER) is returned if a number exceeding the maximum number of image data being able to be registered is set.

If NULL is selected for all identification data, the error (ERR_PARAM) is returned.

If registered data that match all the selected identification data are already present, the error (ERR_EXIST) is returned. Capital and small letters are not distinguished for the arguments of pFileID and pImageTagData. If data that is the same except for the difference of capital and small letters is already present, the error (ERR_EXIST) is returned.

If cropped image data cannot be registered because system memory is lacking or remaining memory on the hard disk is not enough, the error (ERR_NO_MEMORY/ERR_DISK_FULL) is returned.

When saving in a file, if a file that has the same file name is already present, it is overwritten (the content of the original data is not guaranteed.)

If image data reading cannot be executed, the error (ERR_NO_IMAGE) is returned.

When executing BiCloseMonPrinter, all the data defined in Crop image saving table are deleted.

If the format of image data that is read from the printer is raster format (EPS_BI_SCN_RASTER(2)), the non-supported error (ERR_WORKAREA_UNKNOWNFORMAT) is returned.

BiESCNRetrieveImage

Acquire Crop image.

Syntax

nErr = **BiESCNRetrieveImage** (*nHandle , dwFileIndex , pFileID , pImageTagData ,*
pImageSize , pImageData)

Argument

nHandle: Specifies the handle. This is an INT type.

dwFileIndex: Select FileIndex that is identification data of the Crop image to be acquired. If NULL is selected, searching is not executed with this identification data. This is a DWORD type.

pFileID: Select FileID that is identification data of the Crop image to be acquired. If NULL is selected, searching is not executed with this identification data. The symbols \ / : , ; * ? " < > and | cannot be used. This is a LPSTR type.

pImageTagData: Select ImageTagData that is identification data of the Crop image to be acquired. If NULL is selected, searching is not executed with this identification data. The symbols \ / : , ; * ? " < > and | cannot be used. This is a LPSTR type.

pImageSize: Select memory size that sets Crop image data. After executing this argument, the actual CROP image data size is set. If buffer capacity is not enough, necessary bytes are set; however, in the other error cases or if data is not present, nothing is set. This is a LPDWORD type.

pImageData: Select a memory address that sets Crop image data. This is a LPBYTE type.

Return value

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_BUFFER_OVER_FLOW	-140	Lack of buffer capacity
ERR_NOT_FOUND	-220	Selected data is not present
ERR_IMAGE_FILEOPEN	-230	Failure in opening image data file
ERR_IMAGE_FILEREAD	-290	Failure in reading image data file
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed.
ERR_EXEC_SCAN	-330	ERR_EXEC_SCAN Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Acquire Crop image data that match the selected identification data from Crop image save table of status API.

If NULL is selected for all identification data, the error (ERR_PARAM) is returned.

If Crop image data that match identification data selected are not present, the error (ERR_NOT_FOUND) is returned.

If the size of Crop image data is bigger than the memory size selected with pImageSize, the error (ERR_BUFFER_OVER_FLOW) is returned.

If Crop image data that match the selected identification data are present more than once, only the data that has been searched first is acquired.

Example:

If the Crop image data are saved as follows:

(1) When “dwFileIndex=1,pFileID=NULL,pImageTagData=NULL” is specified, fileIndex=1,fileID=“A“ is acquired.

(2) When “dwFileIndex=2,pFileID=NULL,pImageTagData=NULL” is specified, fileIndex=2,fileID=“A“ is acquired.

(3) When “dwFileIndex=NULL,pFileID=”B” ,pImageTagData=NULL” is specified, fileIndex=1,fileID=”B” is acquired.

In order of registration

dwFileIndex	pFileID	pImageTagData	ImageData	
1	“A”	NULL	······	[1]
1	“B”	NULL	······	[3]
1	**“C”**	NULL	······	
2	“A”	NULL	······	[2]
2	“B”	NULL	······	
2	**“C”**	NULL	······	
3	“A”	“A”	······	
3	“B“	NULL	······	

BiESCNClearImage

Clear the registered Crop image.

Syntax

nErr = **BiESCNClearImage** (*nHandle , bFlag , dwFileIndex , pFileID , pImageTagData*)

Argument

nHandle: Specifies the handle. This is an INT type.

bFlag: Select the flag for the method of clearing. The Crop image to be cleared can be identified by a combination of flags. This is a BYTE type.

Macro Definition (Constant)	Value	Description
CROP_CLEAR_ALL_IMAGE	0	Clear all the Crop image data saved
CROP_CLEAR_BY_FILEINDEX	1	Clear all the Crop image data that have the values selected to dwFileIndex
CROP_CLEAR_BY_FILEID	2	Clear all the Crop image data that have the values selected to pFileID
CROP_CLEAR_BY _IMAGETAGDATA	4	Clear all the Crop image data that have the values selected to pImageTagData

Example of selecting:

When CROP_ CLEAR_ BY_FILEINDEX + CROP_ CLEAR_ BY_FILEID + CROP_ CLEAR_ BY_IMAGETAGDATA is selected to bFlag, all the Crop image data that match all the identification data are cleared.

dwFileIndex: Select FileIndex, the identification data of the Crop image to be cleared. This is a DWORD type.

pFileID: Select FileID, the identification data of the Crop image to be cleared. The symbols \ / : , ; * ? "< > and | cannot be used. This is a LPSTR type.

pImageTagData: Select ImageTagData, the identification data of the Crop image to be cleared. The symbols \ / : , ; * ? " < > and | cannot be used. This is a LPSTR type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_NOT_FOUND	-220	Selected data is not present
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	ERR_EXEC_SCAN Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Explanation

Clear the Crop image data that match the selected identification data from the saving table of Crop image data of status API. It is possible to clear all by selecting bFlag.

Note

The error (ERR_NOT_FOUND) is returned if the Crop image data that match the selected identification data are not present.

When executing BiCloseMonPrinter, all the data defined in the saving table of the Crop image are cleared.

BiESCNGetRemainingImages

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

Syntax

nErr = **BiESCNGetRemainingImages** (*nHandle* , *pRemainingImages*)

Argument

nHandle: Specifies the handle. This is an INT type.

pRemainingImages: Select a memory address to set a value of the number of remaining data that can be registered. This is a LPBYTE type.

Return Values

SUCCESS(0) is returned if the execution is successful. If an error occurs, one of the following error codes (negative value) is returned:

Macro Definition (Constant)	Value	Description
SUCCESS	0	SUCCESS
ERR_HANDLE	-60	The handle value that selects the printer is incorrect
ERR_PARAM	-90	Parameter error
ERR_NOT_SUPPORT	-100	Not supported
ERR_EXEC_FUNCTION	-310	Cannot be used because the other API is being executed
ERR_EXEC_SCAN	-330	ERR_EXEC_SCAN Cannot call as the scanner is scanning

NOTE

For information on remedies for the Status API execution errors, refer to "Status API Execution Error" on page 27.

Reference for .NET

This chapter explains Status API and the syntax used in .NET environment.

Properties

AutoPowerOffTime

Calls BiAutoPowerOffTime in APD4StatusAPI and acquires or sets the auto power off time.

Access:	Read only
Data type:	int

Explanation

For details, see "Reference for Win32" on page 41. Processing mode is internally set with Get/Set. For details, see "BiAutoPowerOffTime" on page 106.

ESCNAutoSize

Calls BiESCNGetAutoSize/ BiESCNSetAutoSize in APD4StatusAPI and acquires or sets the AutoSize processing flag value.

Access:	Readable and writable
Data type:	com.epson.pos.driver.AutoSize

Explanation

Constants defined in com.epson.pos.driver.AutoSize shall be used for the value. For details, see "Reference for Win32" on page 41.

ESCNCutSize

Calls BiESCNSetCutSize/ BiESCNSetAutoSize in APD4StatusAPI and acquires the values of or sets values for the size of cutting the right and left sides of the image data.

Access:	Readable and writable
Data type:	int

Explanation

For details, see "Reference for Win32" on page 41.

ESCNDocumentSize

Calls BiESCNGetDocumentSize/ BiESCNSetDocumentSize in APD4StatusAPI and acquires the values of or sets values for the width and height of the image data stored in WORK AREA (unit: 0.1mm).

Access:	Readable and writable
Data type:	System.Drawing.Size

Explanation

The values shall be processed by using a System.Drawing.Size class. In the System.Drawing.Size class, the Width property corresponds to DocumentWidth and the Height property corresponds to DocumentHeight.
For details, see "Reference for Win32" on page 41.

ESCNRemainingImages

Calls BiESCNGetRemainingImages in APD4StatusAPI and acquires the number of the remaining data for the Crop image that can be registered.

Access:	Read only
Data type:	int

Explanation

For details, see "Reference for Win32" on page 41.

ESCNRotate

Calls BiESCNGetRotate/ BiESCNSetRotateRotate in APD4StatusAPI and acquires or sets the processing flag value.

Access:	Readable and writable
Data type:	com.epson.pos.driver.Rotate

Explanation

Constants defined in com.epson.pos.driver.Rotate shall be used for the value. For details, see "Reference for Win32" on page 41.

InkStatus

Calls BiGetInkStatus in APD4StatusAPI and acquires the current ink status.

Access:	Read only
Data type:	com.epson.pos.driver.InkASB

Explanation

Constants defined in com.epson.pos.driver.InkASB shall be used for the value, see "BiGetInkStatus" on page 88.

IsValid

Acquires the open status of the printer.

Access:	Read only
Data type:	System.Boolean

Explanation

Returns either of the following values.

true:	Successfully opened.
false:	Not opened or failed to be opened.

LastError

Acquires the error code of the last executed API.

Access:	Read only
Data type:	com.epson.pos.driver.ErrorCode

Explanation

Can acquire an error code at any time because this module retains the last executed API.
This method is used to judge success or failure because APIs implemented in properties cannot return error codes.

Error codes for errors that may occur in all APIs. For details, see "Status API Execution Error" on page 27.

OfflineCode

Calls BiGetOfflineCode in APD4StatusAPI and acquires the bit that indicates the cause of the printer being offline.

Access:	Read only
Data type:	System.Byte[]

Explanation

For details, see "BiGetOfflineCode" on page 78.

Status

Calls BiGetSatus in APD4StatusAPI and acquires the current printer status.

Access:	Read only
Data type:	com.epson.pos.driver.ASB

Explanation

Constants defined in com.epson.pos.driver.ASB shall be used for the value. For details, see "BiGetStatus" on page 79.

Methods

OpenMonPrinter

Starts controlling the specified printer.

Calls BiOpenMonPrinter in APD4 Status API. For details, see "BiOpenMonPrinter" on page 59.

Prototype

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

Parameters

OpenType type:	Type of name to be specified for name. Constants defined in com.epson.pos.driver.OpenType shall be used for the value.
String name:	Starts controlling the specified printer.

Return value other than APD4 Status API return code

This method returns only an error code defined in com.epson.pos.driver.ErrorCode. (It does not return a handle.)

CloseMonPrinter

Stops controlling the specified printer.

Calls BiCloseMonPrinter in APD4 Status API. For details, see "BiCloseMonPrinter" on page 61.

Prototype

ErrorCode ***CloseMonPrinter***()

LockPrinter

Occupies the printer.

Calls BiLockPrinter in APD4 Status API. For details, see "BiLockPrinter" on page 62.

Prototype

ErrorCode ***LockPrinter***(int timeout)

Parameters

int timeout:	Timeout time (in ms units).

UnlockPrinter

Stops occupying the printer.

Calls BiUnLockPrinter in APD4 Status API. For details, see "BiUnlockPrinter" on page 64.

Prototype

ErrorCode ***UnlockPrinter***()

SetMonInterval

Specifies the time interval for Status Monitor API to read status information from the printer, in ms units.

Calls BiSetMonInterval in APD4 Status API. For details, see "BiSetMonInterval" on page 65.

Prototype

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

Parameters

int noPrnInterval:	Unused
int prnInterval:	Printer monitoring interval

SetMonEtherInterval

Specifies the time interval for Status Monitor API to read status information from the printer on the network, in ms units.

Calls BiSetMonEtherInterval in APD4 Status API. For details, see "BiSetMonEtherInterval" on page 66.

Prototype

ErrorCode ***SetMonEtherInterval***(int EtherInterval)

Parameters

int EtherInterval:	Network printer monitoring interval

DirectIOEx

After sending the specified data to the printer, receives data of the specified length from the printer.

Calls BiDirectIOEx in APD4 Status API. For details, see "BiDirectIOEx" on page 69.

Prototype

ErrorCode ***DirectIOEx***(byte() writeCmd, ref byte() readBuff, int timeout, bool nullTerminate, byte option)

ErrorCode ***DirectIOEx***(byte() writeCmd, out String response, int timeout, byte option)

ErrorCode ***DirectIOEx***(byte() writeCmd, int timeout)

Parameters

byte[] writeCmd:	Data to be sent to the printer
ref byte[] readBuff:	Data received from the printer
int timeout:	Timeout time for data transmission and reception (in ms units)
bool nullTerminate:	Whether or not to terminate reception when NULL is received
byte option:	In the case of "True", reading is complete when NULL is received from the printer. At this time, specify the readBuff size to readLen. In the case of "FALSE ", the length of data specified in readLen is read or data is read from the printer until a timeout error is generated.
out String response:	Data received from the printer (to be converted into strings)

ResetPrinter

Resets the printer. When resetting the printer during printing, cancels print jobs and performs printer resetting.

Calls BiResetPrinter in APD4 Status API. For details, see "BiResetPrinter" on page 73.

Prototype

ErrorCode ***ResetPrinter***()

CancelError

Calls BiCancelError in APD4 Status API. Performs recovery from a printer recoverable error.

For details, see "BiCancelError" on page 75.

Prototype

ErrorCode ***CancelError***()

GetType

Acquires the type ID of the printer. Some information cannot be acquired depending on the model. In such a case, “0”is set.

Calls BiGetType in APD4 Status API. For details, see "BiGetType" on page 77.

Prototype

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

Parameters

out byte typeid:	Type ID of the printer
out byte font:	Fonts installed in the printer
out byte exrom:	Capacity of the printer’s extended Flash ROM.
out byte euspecial:	Special ID of the printer

GetRealStatus

Acquires the most recent status of the printer. Individual bits in the status correspond to the contents of the ASB status and constants defined in com.epson.pos.driver. ASB shall be used.

Calls BiGetRealStatus in APD4 Status API. For details, see "BiGetRealStatus" on page 80.

Prototype

ErrorCode ***GetRealStatus***(out ASB asb)

Parameters

out ASB asb: Current printer status

SetStatusBack

Starts status notification through StatusCallback/StatusCallbackEx events.

Calls BiSetStatusBackFunctionEx in APD4 Status API. For details, see "BiSetStatusBackFunctionEx" on page 84.

Prototype

ErrorCode ***SetStatusBack***()

CancelStatusBack

Stops status notification through StatusCallback/StatusCallbackEx events.

Calls BiCancelStatusBack in APD4 Status API. For details, see "BiCancelStatusBack" on page 87.

Prototype

ErrorCode ***CancelStatusBack***()

ErrorCode ***CancelStatusBack***()

SetInkStatusBack

Starts status notification through InkStatusCallback/InkStatusCallbackEx events.

Calls BiSetInkStatusBackFunctionEx in APD4 Status API. For details, see "BiSetInkStatusBackFunctionEx" on page 91.

Prototype

ErrorCode ***SetInkStatusBack***()

CancelInkStatusBack

Stops status notification through InkStatusCallback/InkStatusCallbackEx events.

Calls BiCancelInkStatusBack in APD4 Status API. For details, see "BiSetInkStatusBackFunction" on page 89.

Prototype

ErrorCode ***CancelInkStatusBack***()

GetBatteryStatus

Acquires the current printer’s battery status.

Calls BiGetBatteryStatus in APD4 Status API. For details, see "BiGetBatteryStatus" on page 95.

Prototype

```
ErrorCode GetBatteryStatus( ref PowerStatus lpbPowerStatus,
                            ref BatteryStatus lpbBatteryStatus )
```

Parameters

ref PowerStatus lpbPowerStatus:	Power statusConstants defined in com.epson.pos.driver.PowerStatus shall be used for the value.
ref BatteryStatus lpbBatteryStatus:	Battery levelConstants defined in com.epson.pos.driver.BatteryStatus shall be used for the value.

SetBatteryStatusBack

Starts battery status notification through a BatteryStatusCallbackEx event.

Calls BatteryStatusCallbackEx in APD4 Status API. For details, see "BiSetBatteryStatusBackFunctionEx" on page 97.

Prototype

ErrorCode ***SetBatteryStatusBack***()

CancelBatteryStatusBack

Stops battery status notification through a BatteryStatusCallbackEx event.

Calls BiCancelBatteryStatusBack in APD4 Status API. For details, see "BiCancelBatteryStatusBack" on page 100.

Prototype

ErrorCode ***CancelBatteryStatusBack***()

SetPresenterEvent

Starts presenter event notification through PresenterCallback/PresenterCallbackEx events.

Calls BiSetPresenterEventBackFunctionEx in APD4 Status API. For details, see "BiSetPresenterEventBackFunctionEx" on page 103.

Prototype

ErrorCode ***SetPresenterEvent***()

CancelPresenterEvent

Stops presenter event notification through PresenterCallback/PresenterCallbackEx events.

Calls CancelPresenterEvent in APD4 Status API. For details, see "BiCancelPresenterEventBack" on page 105.

Prototype

ErrorCode ***CancelPresenterEvent***()

PowerOff

Executes the power-off process of the printer.

Calls BiPowerOff in APD4 Status API. For details, see "BiPowerOff" on page 108.

Prototype

ErrorCode ***PowerOff***()

EjectSheet

Instructs the presenter to eject receipt paper.

Calls BiEjectSheet in APD4 Status API. For details, see "BiEjectSheet" on page 110.

Prototype

ErrorCode ***EjectSheet***()

GetCounter

Reads the maintenance counter.

Calls BiGetCounter in APD4 Status API. For details, see "BiGetCounter" on page 111.

Prototype

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

ErrorCode ***GetCounter***(byte counter, out int value)

Parameters

CounterIndex counter:	Maintenance counter numberConstants defined in com.epson.pos.driver.CounterIndex shall be used for the value.
bool cumulative:	Whether or not the maintenance counter number specified by counter refers to the cumulative counter true: cumulative counter false: reset counter
out int value:	Maintenance counter value
byte counter:	Maintenance counter number

ResetCounter

Resets the maintenance counter.

Calls BiResetCounter in APD4 Status API. For details, see "BiResetCounter" on page 113.

Prototype

ErrorCode ***ResetCounter***(CounterIndex counter)

ErrorCode ***ResetCounter***(byte counter)

Parameters

CounterIndex counter:	Maintenance counter numberConstants defined in com.epson.pos.driver.CounterIndex shall be used for the value.
byte counter:	Maintenance counter number

GetPrnCapability

Acquires information about the printer specified by the printer ID.

Calls BiGetPrnCapability in APD4 Status API. For details, see "BiGetPrnCapability" on page 115.

Prototype

ErrorCode ***GetPrnCapability***(byte printerID, out byte() data)

ErrorCode ***GetPrnCapability***(byte printerID, out String data)

Parameters

byte printerID:	ID of the printer from which information is acquired.
out byte[] data:	Printer information
out String data:	Printer information

OpenDrawer

Activates the drawer. Can be used also when the printer is offline.

Calls BiOpenDrawer in APD4 Status API. For details, see "BiOpenDrawer" on page 117.

Prototype

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

Parameters

Drawer drawer:	Drawer to be openedConstants defined in com.epson.pos.driver.Drawer shall be used for the value.
Pulse pulse:	Interval up to activation of the drawerConstants defined in com.epson.pos.driver. Pulse shall be used for the value.

ErrorCode ***PowerOff***()

PreparePaperChange

Ejects paper in a backfeed direction to prepare for roll paper change.

Calls BiPreparePaperChange in APD4 Status API. For details, see "BiPreparePaperChange" on page 119.

Prototype

ErrorCode ***PreparePaperChange***()

RetractSheet

Instructs the presenter to retract receipt paper.

Calls BiRetractSheet in APD4 Status API. For details, see "BiRetractSheet" on page 120.

Prototype

ErrorCode ***RetractSheet***()

SendDataFile

Registers commands by using the command definition file. For the file format of the command definition file, see the descriptions in the subsequent sections.

Prototype

ErrorCode ***SendDataFile***(String filename)

Parameters

String filename:	Command definition fileFile in the current folder is used if no path is specified.

Explanation

The registered command data is discarded when CloseMonPriner is executed. If the same command name is found to be already registered, aborts command registration and returns an error.

The number of commands that can be registered is restricted only by the available memory space of the system.

Calls BiSendDataFile in APD4 Status API.

For details, see "BiSendDataFile" on page 121.

DirectSendRead

Transmits the commands registered through SendDataFile and receives the data specified with the data type name of data to be received.

Calls BiDirectSendRead in APD4 Status API. For details, see "BiDirectSendRead" on page 123.

Prototype

```
ErrorCode DirectSendRead( String cmdName, String readName,
                          ref byte() readBuf, int timeout, bool
                          nullTerminate )
ErrorCode DirectSendRead( String cmdName, String readName,
                          out String response, int timeout )
```

Parameters

String cmdName:	Command name
String readName:	Data type name of data to be received
ref byte[] readBuf:	Received data
int timeout:	Timeout time for data transmission and data reception (in ms units)
bool nullTerminate:	Whether or not to terminate reception when NULL is received
out String response:	Received data

SetDefaultEchoTime

Sets initial values for the number of response confirmation times and for the timeout time per one response confirmation, to the network printer.
When APD4StatusAPI is installed for the first time, the number of confirmation response times is one and the timeout time per one response confirmation is 1 second.
The values set will be enabled the next time OpenMonPrinter is called.
Note that this API is executable only during Ethernet connection.

Calls BiSetDefaultEchoTime in APD4 Status API. For details, see "BiSetDefaultEchoTime" on page 126.

Prototype

ErrorCode ***SetDefaultEchoTime***(int count, int timeout)

Parameters

int count:	Number of response confirmation times
int timeout:	Timeout time per one response confirmation (in ms units)

SetEtherEchoTime

Sets values for the number of response confirmation times and for the timeout time per one response confirmation, to the network printer. Before this API executes, the values set with SetDefaultEchoTime are used. The values set are valid until OpenMonPrinter is called. Note that this API is enabled only during Ethernet connection.

Calls BiSetEtherEchoTime in APD4 Status API. For details, see "BiSetEtherEchoTime" on page 127.

Prototype

ErrorCode ***SetEtherEchoTime***(int count, int timeout)

Parameters

int count:	Number of response confirmation times
int timeout:	Timeout time per one response confirmation

MICRCleaning

Cleans the MICR mechanism. After cleaning is complete, roll paper is selected as a printing sheet.

Calls BiMICRCleaning in APD4 Status API. For details, see "BiMICRCleaning" on page 129.

Prototype

ErrorCode ***MICRCleaning***()

MICRGetStatus

Acquires the MICR status.

Calls BiMICRGetStatus in APD4 Status API. For details, see "BiMICRGetStatus" on page 130.

Prototype

ErrorCode ***MICRGetStatus***(out byte status)

Parameters

out byte status:	MICR status

MICRLoadCheck

Loads the check slip into the print start position.

Calls BiMICRLoadCheck in APD4 Status API. For details, see "BiMICRLoadCheck" on page 132.

Prototype

ErrorCode ***MICRLoadCheck***()

MICRReadCheck

Reads the check slip.

Calls BiMICRReadCheck in APD4 Status API. For details, see "BiMICRReadCheck" on page 133.

Prototype

ErrorCode ***MICRReadCheck***(MicrFont readFont, int waitInsertionTime)

Parameters

MicrFont readFont:	Read fontConstants defined in com.epson.pos.driver.MicrFont shall be used for the value.
int waitInsertionTime:	Check slip insertion wait timeIf 255 (the hidden feature) is specified, nothing is specified for the slip insertion wait time.

MICREjectCheck

Ejects the check slip.

Calls BiMICREjectCheck in APD4 Status API. For details, see "BiMICREjectCheck" on page 135.

Prototype

ErrorCode ***MICREjectCheck***()

MICRCancelWaitCheckInsertion

Cancels the check slip insertion wait status. Also when called by a printer without MICR, cancels the slip insertion wait status.

Calls BiMICRCancelWaitCheckInsertion in APD4 Status API. For details, see "BiMICRCancelWaitCheckInsertion" on page 136.

Prototype

```
ErrorCode MICRCancelWaitCheckInsertion()
```

MICRRetransmissionCheckData

Executes retransmission of the results of check slip reading.

Calls BiMICRRetransmissionCheckData in APD4 Status API. For details, see "BiMICRRetransmissionCheckData" on page 137.

Prototype

```
ErrorCode MICRRetransmissionCheckData
                         ( out String response, out MICRStatus status,
                           out MICRDetail detail, int timeout )
```

Parameters

out String response:	Reading data
out MICRStatus status:	Reading status
out MICRDetail detail:	Reading status detailed information
int timeout:	Timeout time (in ms units)

MICRSelectDataHandling

Configures the operation of check slip reading.

Calls BiMICRSelectDataHandling in APD4 Status API. For details, see "BiMICRSelectDataHandling" on page 139.

Prototype

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

Parameters

CharSelect charSel: How to handle characters that cannot be analyzedConstants defined in com.epson.pos.driver.CharSelect shall be used for the value.

DetailSelect detailSel: Availability of detailed information at the time of abnormal termination of reading Constants defined in com.epson.pos.driver.DetailSelect shall be used for the value.

ErrorSelect errorSel: Whether or not to end the MICR function at the time of abnormal termination. Regardless of this parameter setting, the MICR function shall continue at the time of normal termination or at the time of abnormal termination adding reading results. Constants defined in com.epson.pos.driver.ErrorSelect shall be used for the value.

MICRSetReadBack

Starts MICR data reception notification through MICRReadCallback/MICRReadCallbackEx events.

Calls BiMICRSetReadBackFunctionEx in APD4 Status API. For details, see "BiMICRSetReadBackFunctionEx" on page 144.

Prototype

```
ErrorCode MICRSetReadBack()
```

MICRCancelReadBack

Stops MICR data reception notification through MICRReadCallback/MICRReadCallbackEx events.

Calls BiMICRCancelReadBack in APD4 Status API. For details, see "BiMICRCancelReadBack" on page 148.

Prototype

ErrorCode ***MICRCancelReadBack***()

SCNClumpPaper

Sets the single sheet into the clump state. After this API executes, the printer is transitioned to the single sheet insertion wait status and the sheet is clumped after being inserted. In the above case, if another sheet is already clumped, nothing is done.

Calls BiSCNClumpPaper in APD4 Status API. For details, see "BiSCNClumpPaper" on page 149.

Prototype

ErrorCode ***SCNClumpPaper***()

SCNGetClumpStatus

Acquires the clump status of the single sheet.

Calls BiSCNGetClumpStatus in APD4 Status API. For details, see "BiSCNGetClumpStatus" on page 150.

Prototype

ErrorCode ***SCNGetClumpStatus***(out byte status)

Parameters

out byte status:	Single sheet clump status

SCNGetCroppingAreas

Acquires Cropping area information from the printer.

Calls BiSCNGetCroppingArea in APD4 Status API. For details, see "BiSCNGetCroppingArea" on page 152.

Prototype

ErrorCode ***SCNGetCroppingAreas***(out byte() areas)

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

Parameters

out byte[] areas:	Cropping area information
out Hashtable areas:	Cropping area information

SCNSetCroppingArea

Sets a cropping area. Note that the values set with this API are valid until the powering off or resetting of the printer.

Calls BiSCNSetCroppingArea in APD4 Status API. For details, see "BiSCNSetCroppingArea" on page 154.

Prototype

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

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

Parameters

byte areaNo:	Cropping area number
int startX:	Starting X coordinate of the cropping area
int startY:	Starting Y coordinate of the cropping area
int endX:	Ending X coordinate of the cropping area
int endY:	Ending Y coordinate of the cropping area
Rectangle area:	Cropping area

SCNDeleteCroppingArea

Deletes a cropping area set to the printer.

Calls BiSCNDeleteCroppingArea in APD4 Status API. For details, see "BiSCNDeleteCroppingArea" on page 156.

Prototype

ErrorCode ***SCNDeleteCroppingArea***(byte areaNo)

Parameters

byte areaNo:	Number of a cropping area to be deletedIf 0 is specified, deletes all cropping areas.

SCNGetImageFormat

Acquires the image data format from the printer.

Calls BiSCNGetImageFormat in APD4 Status API. For details, see "BiSCNGetImageFormat" on page 157.

Prototype

ErrorCode ***SCNGetImageFormat***(out Format format)

Parameters

out Format format:	Image data format

SCNSetImageFormat

Sets the format for the image data to be read. Note that the values set with this API are valid until the powering off or resetting of the printer.

Calls BiSCNSetImageFormat in APD4 Status API. For details, see "BiSCNSetImageFormat" on page 159.

Prototype

ErrorCode ***SCNSetImageFormat***(Format format)

Parameters

Format format: Image data formatThe initial value after powering on or resetting the printer is EPS_BI_SCN_TIFF.

SCNGetImageList

Acquires data IDs for the image scanning results that exist in the printer's NV memory for storage.

Calls BiSCNGetImageList in APD4 Status API. For details, see "BiSCNGetImageList" on page 161.

Prototype

ErrorCode ***SCNGetImageList***(out int() idList)

Parameters

out int[] idList: Data IDs held by the printer

SCNGetImageQuality

Acquires the image scanning quality from the printer.

Calls BiGetImageQuality in APD4 Status API. For details, see "BiSCNGetImageQuality" on page 163.

Prototype

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

Parameters

out ColorDepth colorDepth:	Tonal gradation (the number of bits per pixel) Constants defined in com.epson.pos.driver.ColorDepth shall be used for the value.
out double threshold:	Density threshold
out Color color:	Color Constants defined in com.epson.pos.driver.Color shall be used for the value.
out ExOption exOption:	Density adjustment typeConstants defined in com.epson.pos.driver.ExOption shall be used for the value.

SCNSetImageQuality

Sets the image scanning quality. Note that the values set with this API are valid until the powering off or resetting of the printer.

Calls BiSCNSetImageQuality in APD4 Status API. For details, see "BiSCNSetImageQuality" on page 165.

Prototype

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

Parameters

ColorDepth colorDepth: Tonal gradation (the number of bits per pixel) Constants defined in com.epson.pos.driver.ColorDepth shall be used for the value.

double threshold: Density threshold

Color color: Color
Constants defined in com.epson.pos.driver.Color shall be used for the value.

ExOption exOption: Density adjustment type
Constants defined in com.epson.pos.driver.ExOption shall be used for the value.

SCNGetImageRemainingCapacity

Acquires the capacity of the unused area in the NV memory for storage, from the printer.

Calls BiSCNGetImageRemainingCapacity in APD4 Status API. For details, see "BiSCNGetImageRemainingCapacity" on page 167.

Prototype

```
ErrorCode SCNGetImageRemainingCapacity( out int size )
```

Parameters

out int size: Remaining capacity of the printer’s NV memory for storage

SCNGetScanArea

Acquires the image scan area.

Calls BiSCNGetScanArea in APD4 Status API. For details, see "BiSCNGetScanArea" on page 169.

Prototype

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

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

Parameters

out Rectangle area:	Scan area
out int startX:	Starting X coordinate of the scan area
out int startY:	Starting Y coordinate of the scan area
out int endX:	Ending X coordinate of the scan area
out int endY:	Ending Y coordinate of the scan area

SCNSetScanArea

Sets an image scan area.

Calls BiSCNSetScanArea in APD4 Status API. For details, see "BiSCNSetScanArea" on page 171.

Prototype

ErrorCode ***SCNSetScanArea***(Rectangle area)

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

Parameters

Rectangle area:	Scan area
int startX:	Starting X coordinate of the scan area
int startY:	Starting Y coordinate of the scan area
int endX:	Ending X coordinate of the scan area
int endY:	Ending Y coordinate of the scan area

SCNPreScan

After an attempt to scan an image, sets the printer's density threshold and acquires the density threshold set.

Calls BiSCNPreScan in APD4 Status API. For details, see "BiSCNPreScan" on page 174.

Prototype

ErrorCode ***SCNPreScan***(out PreScanResult result, out double threshold)

Parameters

out PreScanResult result: Prescan results status

out double threshold: Density threshold for print paper

SCNReadImage

Executes image scanning.

Calls BiSCNReadImage in APD4 Status API. For details, see "BiSCNReadImage" on page 176.

Prototype

ErrorCode ***SCNReadImage***
(int id, SelectSheet selectSheet, byte waitInsertionTime, byte() addInforData, NVMemory memory)

Parameters

int id: Data identifier to be associated with a scanning image

SelectSheet selectSheet: Scan target paperConstants defined in com.epson.pos.driver.SelectSheet shall be used for the value.

byte waitInsertionTime: Paper insertion wait time (in ms units)

byte[] addInforData: Character dataSpecified when character data is added to an image.

NVMemory memory: Whether or not to save the results of image scanning into the NV memory for storage Constants defined in com.epson.pos.driver. NVMemory shall be used for the value.

SCNRetransmissionImage

Executes retransmission of the results of image scanning.

Calls BiSCNRetransmissionImage in APD4 Status API. For details, see "BiSCNRetransmissionImage" on page 179.

Prototype

```
ErrorCode SCNRetransmissionImage
                ( int id, out byte() data, out int imageX, out
                  SCNStatus status, out SCNDetail detail, int timeout
                )
```

Parameters

int id:	Data identifier to be associated with a scanning image
out byte[] data:	Image data
out int imageX:	Number of image data (bytes) in the X direction
out SCNStatus status:	Scan status Constants defined in com.epson.pos.driver.SCNStatus shall be used for the value.
out SCNDetail detail:	Scan status detailed information Constants defined in com.epson.pos.driver.SCNDetail shall be used for the value.
int timeout:	Timeout time for data reading (in ms units)

SCNDeleteImage

Deletes the image scanning results of data whose data ID is specified.

Calls BiSCNDeleteImage in APD4 Status API. For details, see "BiSCNDeleteImage" on page 181.

Prototype

```
ErrorCode SCNDeleteImage( int id )
```

Parameters

int id:	Data ID of data to be deleted

SCNSelectScanUnit

Sets a unit that executes image scanning. Before execution of this API, the default check paper scan unit (PS_BI_SCN_UNIT_CHECKPAPER) is set.

Calls BiSelectScanUnit in APD4 Status API. For details, see "BiSCNSelectScanUnit" on page 183.

Prototype

ErrorCode ***SCNSelectScanUnit***(ScanUnit scanUnit)

Parameters

ScanUnit scanUnit: Scan unitConstants defined in com.epson.pos.driver.ScanUnit shall be used for the value.

SCNSetReadBack

Starts scan data reception notification through SCNReadCallback/SCNReadCallbackEx events.

Calls BiSCNSetReadBackFunctionEx in APD4 Status API. For details, see "BiSCNSetReadBackFunctionEx" on page 188.

Prototype

ErrorCode ***SCNSetReadBack***()

SCNCancelReadBack

Stops scan data reception notification through SCNReadCallback/SCNReadCallbackEx events.

Calls BiSCNCancelReadBack in APD4 Status API. For details, see "BiSCNCancelReadBack" on page 192.

Prototype

ErrorCode ***SCNCancelReadBack***()

ESCNEnable

Enables the scanner extended functions.

Calls BiESCNEnable in APD4 Status API. For details, see "BiESCNEnable" on page 193.

Prototype

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

Parameters

Storage storage:	Crop image storage method Constants defined in com.epson.pos.driver.Storage shall be used for the value.

ESCNDefineCropArea

Registers Crop information, by using the specified CropAreaID, into the CropArea definition table of Status API.

Calls BiESCNDefineCropArea in APD4 Status API. For details, see "BiESCNDefineCropArea" on page 202.

Prototype

```
ErrorCode ESCNDefineCropArea( byte areaNo, int startX,
                              int startY, int endX, int endY )
```

Parameters

byte areaNo:	CropAreaID of the CropArea to be registered
int startX:	Starting X coordinate of the CropArea
int startY:	Starting Y coordinate of the CropArea
int endX:	Ending X coordinate of the CropArea
int endY:	Ending Y coordinate of the CropArea

ESCNGetMaxCropAreas

Acquire a maximum number of data of CropArea that can be registered.

Calls BiESCNGetMaxCropAreas in APD4 Status API. For details, see "BiESCNGetMaxCropAreas" on page 204.

Prototype

ErrorCode ***ESCNGetMaxCropAreas***(out int count)

Parameters

out int count: Maximum number of data of CropArea that can be registered.

ESCNStoreImage

Registers a Crop image.

Calls BiESCNStoreImage in APD4 Status API. For details, see "BiESCNStoreImage" on page 205.

Prototype

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

Parameters

int fileIndex: ID of a Crop image to be registered

String fileID: File ID of a Crop image to be registered
[\ / : , ; * ? “ < > l]cannot be used.

String imageTagData: Image tag data of a Crop image to be registered
[\ / : , ; * ? “ < > l]cannot be used.

byte cropAreaID: CropAreaID already defined

ESCNRetrieveImage

Acquires Crop image data.

Calls BiESCNRetrieveImage in APD4 Status API. For details, see "BiESCNRetrieveImage" on page 208.

Prototype

ErrorCode ***ESCNRetrieveImage***(int fileIndex, String fileID, String imageTagData, out byte() data)

Parameters

int fileIndex:	ID of a Crop image to be acquired
String fileID:	File ID of a Crop image to be acquired
String imageTagData:	Image tag data of a Crop image to be acquired
out byte[] data:	Crop image data

ESCNClearImage

Deletes a registered Crop image.

Calls BiESCNClearImage in APD4 Status API. For details, see "BiESCNClearImage" on page 211.

Prototype

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

Parameters

ClearImageFlag flag:	Delete methodConstants defined in com.epson.pos.driver.ClearImageFlag shall be used for the value.
int fileIndex:	ID of a Crop image to be deleted
String fileID:	File ID of a Crop image to be deleted
String imageTagData:	Image tag data of a Crop image to be deleted

Events

StatusCallback

Event that handles ASB status notification.

Corresponds to the callback function specified by BiSetStatusBackFunction in APD4 Status API.
For details, see "BiSetStatusBackFunction" on page 82.

Prototype

StatusCallbackHandler(ASB asb)

Parameters

ASB asb:	ASB statusConstants defined in com.epson.pos.driver. ASB shall be used for the value.

StatusCallbackEx

Event that handles ASB status notification.

Corresponds to the callback function specified by BiSetStatusBackFunctionEx in APD4 Status API.
For details, see "BiSetStatusBackFunctionEx" on page 84.

Prototype

StatusCallbackHandlerEx(ASB asb, String portName)

Parameters

ASB asb:	ASB statusConstants defined in com.epson.pos.driver. ASB shall be used for the value.
String portName:	Port name

InkStatusCallback

Event that handles ink status notification.

Corresponds to the callback function specified by BiSetInkStatusBackFunction in APD4 Status API. For details, see "BiSetInkStatusBackFunction" on page 89.

Prototype

InkStatusCallback(InkASB ink)

Parameters

InkASB ink:	Ink statusConstants defined in com.epson.pos.driver.InkASB shall be used for the value.

InkStatusCallbackEx

Event that handles ink status notification.

Corresponds to the callback function specified by BiSetInkStatusBackFunctionEx in APD4 Status API. For details, see "BiSetInkStatusBackFunctionEx" on page 91.

Prototype

InkStatusCallbackEx(InkASB ink, String portName)

Parameters

InkASB ink:	Ink statusConstants defined in com.epson.pos.driver.InkASB shall be used for the value.
String portName:	Port name

BatteryStatusCallbackEx

Event that handles power status notification.

Corresponds to the callback function specified by BiSetBatteryStatusBackFunctionEx in APD4 Status API. For details, see "BiSetBatteryStatusBackFunctionEx" on page 97.

Prototype

```
BatteryStatusCallbackEx( byte bPowerStatus,
                         byte bBatteryStatus,
                         String lpcPortName )
```

Parameters

byte bPowerStatus:	Power status
byte bBatteryStatus:	Battery status
String lpcPortName:	Port name

MICRReadCallback

Event that handles MICR data notification.

Corresponds to the callback function specified by BiMICRSetReadBackFunction in APD4 Status API.

For details, see "BiMICRSetReadBackFunction" on page 141.

Prototype

```
MICRReadCallbackHandler( String data, MICRStatus status,
                         MICRDetail detail )
```

Parameters

String data:	MICR data
MICRStatus status:	Reading status Constants defined in com.epson.pos.driver.MICRStatus shall be used for the value.
MICRDetail detail:	Reading status detailed information Constants defined in com.epson.pos.driver.MICRDetail shall be used for the value.

MICRReadCallbackEx

Event that handles MICR data notification.

Corresponds to the callback function specified by BiMICRSetReadBackFunctionEx in APD4 Status API.
For details, see "BiMICRSetReadBackFunctionEx" on page 144.

Prototype

MICRReadCallbackHandlerEx(String data, MICRStatus status,
MICRDetail detail, String portName)

Parameters

String data: MICR data

MICRStatus status: Reading status Constants defined in com.epson.pos.driver.MICRStatus shall be used for the value.

MICRDetail detail: Reading status detailed information Constants defined in com.epson.pos.driver.MICRDetail shall be used for the value.

String portName: Port name

SCNReadCallback

Event that handles Scan data notification.

Corresponds to the callback function specified by BiSCNSetReadBackFunction in APD4 Status API.
For details, see "BiSCNSetReadBackFunction" on page 185.

Prototype

SCNReadCallbackHandler(byte() data, int imageX,
SCNStatus status, SCNDetail detail)

Parameters

byte[] data: Scan data

int imageX: Number of image data (bytes) in the X direction

SCNStatus status: Scan status Constants defined in com.epson.pos.driver.SCNStatus shall be used for the value.

SCNDetail detail: Scan status detailed information Constants defined in com.epson.pos.driver.SCNDetail shall be used for the value.

SCNReadCallbackEx

Event that handles Scan data notification.

Corresponds to the callback function specified by BiSCNSetReadBackFunctionEx in APD4 Status API. For details, see "BiSCNSetReadBackFunctionEx" on page 188.

Prototype

```
SCNReadCallbackHandlerEx
                   ( byte() data, int imageX, SCNStatus status,
                     SCNDetail detail, String portName )
```

Parameters

byte[] data:	Scan data
int imageX:	Number of image data (bytes) in the X direction
SCNStatus status:	Scan status Constants defined in com.epson.pos.driver.SCNStatus shall be used for the value.
SCNDetail detail:	Scan status detailed information Constants defined in com.epson.pos.driver.SCNDetail shall be used for the value.
String portName:	Port name

PresenterCallback

Event that handles presenter event notification

Corresponds to the callback function specified by BiSetPresenterEventBackFunction in APD4 Status API.
For details, see "BiSetPresenterEventBackFunction" on page 101.

Prototype

PresenterCallbackHandler(byte returncode)

Parameters

byte returncode:	Operating results of the presenter.

PresenterCallbackEx

Event that handles presenter event notification

Corresponds to the callback function specified by BiSetPresenterEventBackFunctionEx in APD4 Status API.
For details, see "BiSetPresenterEventBackFunctionEx" on page 103.

Prototype

PresenterCallbackHandlerEx(byte returncode, String portName)

Parameters

byte returncode:	Operating results of the presenter.
String portName:	Port name

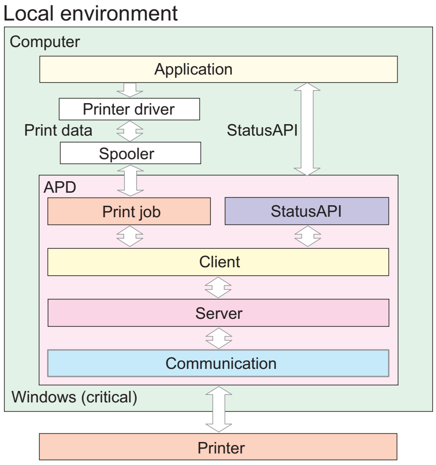
Generating Log Files

This chapter explains how to output and view log files.

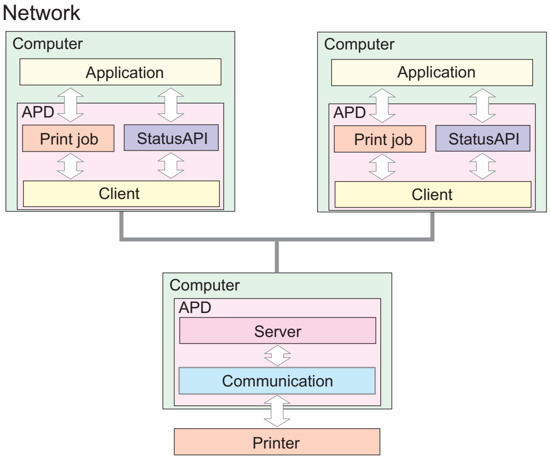
The APD allows you to create a log file which can help you to troubleshoot a problem quickly. The log file is saved under a name of EpsonPOSPort.log. The main features are as follows.

- When a trouble occurs in a printing system, you can view Windows error information in addition to the APD log.
- A process ID can be acquired. You can identify which log is for which process when multiple processes have been executed.
- Client-server system is supported. Logs of client/server module can be acquired.
- Log file can be generated for each of the following items; Status API, print job, client, server, and communication module.

The following diagram shows where the APD log is acquired. Logs for each module are output as a single log file in the order of acquisition.



The module configuration of client-server system is shown below.

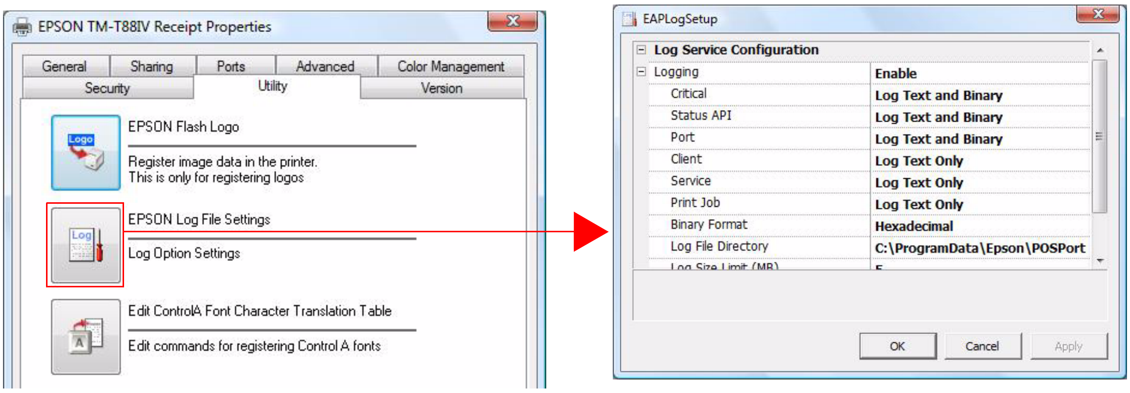


A log file is generated for each computer. The computer's date and time setting is applied to the time stamp.

Log Files Settings

You can select whether to output the log file or not, which module to be logged, and where to output the log file.

Select the Utility tab on the properties screen, and click the [EPSON Log File Settings] button.



EPSON Flash Logo

Register image data in the printer.
This is only for registering logos

EPSON Log File Settings

Log Option Settings

Edit ControlA Font Character Translation Table

Edit commands for registering Control A fonts

EAPLogSetup

Log Service Configuration

Logging	Enable
Critical	Log Text and Binary
Status API	Log Text and Binary
Port	Log Text and Binary
Client	Log Text Only
Service	Log Text Only
Print Job	Log Text Only
Binary Format	Hexadecimal
Log File Directory	C:\ProgramData\Epson\POSPort
Log Size Limit (MB)	5

Make the following settings on the tab.

Setting		Description
Logging	Enable (Default)	Enables log output.
	Disable	Disables log output.
Critical	Select how the Windows error information is output.	
	Log Text Only	Outputs the log as text data.
	Log Text and Binary (Default)	Outputs the log as text and binary data.
Status API	Select how the Status API log is output.	
	Do Not log	A Status API log is not output.
	Log Text Only	Outputs the log as text data.
	Log Text and Binary (Default)	Outputs the log as text and binary data.
Port	Select how the log of the communication port is output.	
	Do Not log	A communications port log is not output.
	Log Text Only	Outputs the log as text data.
	Log Text and Binary (Default)	Outputs the log as text and binary data.
Client	Select how the log of the application on the client-server system is output.	
	Do Not log	A client log is not output.
	Log Text Only (Default)	Outputs the log as text data.
	Log Text and Binary	Outputs the log as text and binary data.
Service	Select how the log of the server on the client-server system is output.	
	Do Not log	A service log is not output.
	Log Text Only (Default)	Outputs the log as text data.
	Log Text and Binary	Outputs the log as text and binary data.

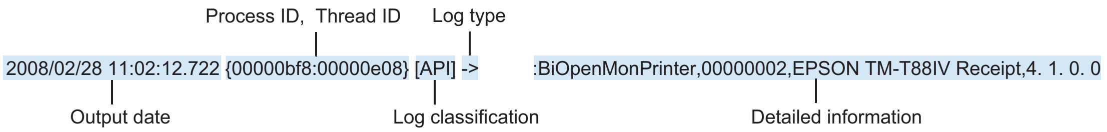
Setting		Description
Print Job	Select how the log of print jobs is output.	
	Do Not log	A print job information log is not output.
	Log Text Only (Default)	Outputs the log as text data.
	Log Text and Binary	Outputs the log as text and binary data.
Binary Format	Configures the binary data format.	
	Hexadecimal (Default)	Outputs the log to a hexadecimal log file.
	Base64	Outputs the log to a Base64 log file.
Log File Directory	Specify where to output the log file. (Default) Windows XP: C:\Documents and Settings\All Users\Application Data\Epson\POSPort Windows 7 / Vista: C:\ProgramData\Epson\POSPort	
Log Size Limit (MB)	Specify the upper limit of the log file size. When the upper limit is exceeded, the log file is compressed using zip format and saved as a BAK file. The subsequent log information is saved as a new log file. A sequential number is added to the name of the BAK files. (example: EpsonPOSPort1.bak). Specify the number of log files to be backed up. (Range: 1 to 1024 , Default: 5)	
Backup File Count	Specify the number of log files to be backed up. (Range 1 to 9 , Default: 1)	

NOTE

Output function of hexadecimal dumping list is not supported.

Viewing Log Files

A log file is viewed as follows.



```
2008/02/28 11:02:12.722 {00000bf8:00000e08} [API] ->        :BiOpenMonPrinter,00000002,EPSON TM-T88IV Receipt,4. 1. 0. 0
```

Log Data	Explanation
Output Date	YYYY/MM/DD hh:mm:ss.sss
Process ID, Thread ID	{Process ID: Thread ID}
Log classification	Shows of which module the log file is. (See "Log classification" on page 260.)
Log type	-> Call function <- Return function -- Execute function (call, return) ** Occurrence of an event

Log Data	Explanation
Detailed information	Information for each module and log type When a function is executed: Function name (parameter 1, ---, parameter n) <return> (execution time (ms))

Log classification

Module	Log classification	Contents of detailed information
Critical	!!!	Important events and errors on Windows
Status API	API	Status API call function and its parameter information, or called Status API function and its parameter and return information
Port	PRT	Port controls, events specific to the interface, and input/output data information
Client	CLI	Process information of the application on the client-server system.
Server	SVR	Process information of the server on the client-server system.
Print Job	SPL	Port open/close information and input/output to/from the port information

Log output example

```
2008/02/28 11:02:12.722 {00000bf8:00000e08} [API] ->          :BiOpenMonPrinter,00000001,EPSON TM-T88IV Receipt,4.
                    1. 2. 0
2008/02/28 11:02:12.722 {00000bf8:00000e08} [CLI] -> Open('pipe://TM/ESDPRT001', 0x01473460)
2008/02/28 11:02:12.722 {000006dc:00000cd0} [SVR] -> 0036d4e8::Open(0, TM/ESDPRT001)
2008/02/28 11:02:12.722 {000006dc:00000cd0} [SVR] <- 0036d4e8::Open(8, TM/ESDPRT004) <00000000>
2008/02/28 11:02:12.722 {00000bf8:00000e08} [CLI] ** (TM/ESDPRT001)Event(0x00010003) 4:
2008/02/28 11:02:12.722 {000006dc:00000cdc} [SVR] -- 0036d4e8::RegisterCallback(8, 00010001) <PHR_SUCCESS>
2008/02/28 11:02:12.722 {00000bf8:00000e08} [CLI] <- Open('pipe://TM/ESDPRT001', 1) <00000000>
2008/02/28 11:02:12.722 {00000bf8:00000e08} [CLI] -> RegisterCallback(1, 0x00040002, 0x01396e00, 0x0177f2f0)
2008/02/28 11:02:12.722 {000006dc:00000c7c} [SVR] -- 0036d4e8::RegisterCallback(8, 00040002) <PHR_SUCCESS>
2008/02/28 11:02:12.722 {00000bf8:00000e08} [CLI] <- RegisterCallback(1, 0x00040002, 0x01396e00, 0x0177f2f0)
                   <00000000>
```

Appendix

Model Information

This document explains the information acquired by Status API for the different printer models.

BA-T500

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_PLATEN_OPEN	ON	0x00000020	Platen is open
	OFF	0x00000000	Platen is closed
ASB_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated

Macro Definitions	ON/ OFF	Value	Status
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

EU-T300

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_PLATEN_OPEN	ON	0x00000020	Platen is open
	OFF	0x00000000	Platen is closed
ASB_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

EU-T400

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_PRESENTER_COVER	ON	0x00000004	Presenter cover is open
	OFF	0x00000000	Presenter cover is closed
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_PLATEN_OPEN	ON	0x00000020	Platen is open
	OFF	0x00000000	Platen is closed
ASB_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
ASB_RECEIPT_NEAR_END_FIRST	ON	0x00020000	Paper in the roll paper near first end detector
	OFF	0x00000000	No paper in the roll paper near first end detector
ASB_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector

Macro Definitions	ON/ OFF	Value	Status
ASB_RECEIPT_NEAR_END_SECOND	ON	0x00400000	Paper in the roll paper near second end detector
	OFF	0x00000000	No paper in the roll paper near second end detector
ASB_PRESENTER_TE	ON	0x01000000	Presenter T/E receipt end
	OFF	0x00000000	Presenter T/E receipt not end
ASB_PRESENTER_TT	ON	0x02000000	Presenter T/T receipt end
	OFF	0x00000000	Presenter T/T receipt not end
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Offline Cause

Code	Status
00h	No offline cause.
20h	Platen is open.
21h	Paper FEED button is pressed.
22h	No paper.
23h	Presenter is open. (not located in the standby position)
40h	Auto cutter error occurs.
41h	Platen open error occurs.(Platen is open during printing)
42h	Black Mark detection error occurs.
43h	Presenter error occurs.
60h	CPU execution error occurs.
61h	Low voltage error occurs. (Lower than the specified power supply voltage)
62h	High voltage error occurs. (Higher than the specified power supply voltage)
63h	Memory read/write error occurs.
64h	Drive circuit connection abnormal error occurs.

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
61 (3DH)	Resetable	Number of paper presenter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
189 (BDH)	Cumulative	Number of paper presenter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

EU-T500

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_PRESENTER_COVER	ON	0x00000004	Presenter cover is open
	OFF	0x00000000	Presenter cover is closed
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_PLATEN_OPEN	ON	0x00000020	Platen is open
	OFF	0x00000000	Platen is closed
ASB_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
ASB_RECEIPT_NEAR_END_FIRST	ON	0x00020000	Paper in the roll paper near first end detector
	OFF	0x00000000	No paper in the roll paper near first end detector
ASB_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector

Macro Definitions	ON/ OFF	Value	Status
ASB_RECEIPT_NEAR_END_SECOND	ON	0x00400000	Paper in the roll paper near second end detector
	OFF	0x00000000	No paper in the roll paper near second end detector
ASB_PRESENTER_TE	ON	0x01000000	Presenter T/E receipt end
	OFF	0x00000000	Presenter T/E receipt not end
ASB_PRESENTER_TT	ON	0x02000000	Presenter T/T receipt end
	OFF	0x00000000	Presenter T/T receipt not end
ASB_RETRACTOR_R1JAM	ON	0x04000000	Presenter receipt end R1JAM
	OFF	0x00000000	-
ASB_RETRACTOR_R2JAM	ON	0x20000000	Retractor receipt end R2JAM
	OFF	0x00000000	-
ASB_RETRACTOR_SENSOR3	ON	0x40000000	Receipt end retractor box
	OFF	0x00000000	-
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Offline Cause

Code	Status
00h	No offline cause.
20h	Platen is open.
21h	Paper FEED button is pressed.
22h	No paper.
23h	Presenter is open. (not located in the standby position)
40h	Auto cutter error occurs.
41h	Platen open error occurs.(Platen is open during printing)
42h	Black Mark detection error occurs.
43h	Presenter error occurs.
60h	CPU execution error occurs.
61h	Low voltage error occurs. (Lower than the specified power supply voltage)
62h	High voltage error occurs. (Higher than the specified power supply voltage).
63h	Memory read/write error occurs.
64h	Drive circuit connection abnormal error occurs.

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
61 (3DH)	Resetable	Number of paper presenter operations	Times
62 (3EH)	Resetable	Number of Paper retractor operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
189 (BDH)	Cumulative	Number of paper presenter operations	Times
190 (BEH)	Cumulative	Number of Paper retractor operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-C100

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_PAPER_SELECTED	ON	0x01000000	Roll paper is selected
	OFF	0x00000000	Cut-sheet is selected
ASB_SLIP_STATUS	ON	0x02000000	Paper status, printing is impossible
	OFF	0x00000000	Paper status, printing is possible
ASB_ASF_SELECTED	ON	0x04000000	ASF selected as the active sheet
	OFF	0x00000000	Cut-sheet selected as the active sheet
ASB_PAPER_INSWAIT	ON	0x40000000	Waiting for cut sheets to be loaded
	OFF	0x00000000	-

Macro Definitions	ON/ OFF	Value	Status
ASB_RELEASE_OPEN	ON	0x40000000	Open the release lever
	OFF	0x00000000	Closes the release lever
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Ink status

Macro Definitions	ON/ OFF	Value	Status
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 cartridge present
INK_ASB_CLEANING	ON	0x0020	Cleaning
	OFF	0x0000	-

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	Overcurrent error generated
	OFF	0x00	Overcurrent error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Thermistor error generated
	OFF	0x00	Thermistor error not generated
1	ON	0x02	Head high voltage error
	OFF	0x00	No head high voltage error
2	ON	0x04	Head low voltage error
	OFF	0x00	No head low voltage error
3	ON	0x08	RTC error
	OFF	0x00	No RTC error
4	ON	0x10	Carriage drive count error
	OFF	0x00	No carriage drive count error
5	ON	0x20	Pump drive count error
	OFF	0x00	No carriage drive count error
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Auto cutter error generated
	OFF	0x00	Auto cutter error not generated
1	-	0x00	Reserved (fixed to 0)
2	-	0x00	Reserved (fixed to 0)
3	-	0x00	Reserved (fixed to 0)
4	ON	0x10	Carriage home position detection error
	OFF	0x00	No carriage home position detection error
5	ON	0x20	Unsynchronized carriage detection error
	OFF	0x00	No unsynchronized carriage detection error
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

Bit	ON/ OFF	Value	Status
0	-	0x00	Reserved (fixed to 0)
1	ON	0x02	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	ON	0x04	Low head temperature error generated
	OFF	0x00	Low head temperature error not generated
3	ON	0x08	Motor drive temperature error
	OFF	0x00	No motor drive temperature error
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
30 (1EH)	Resetable	Number of line feeds (for roll paper)	Lines
31 (1FH)	Resetable	Average number of head shots (color 1)	Shots
32 (20H)	Resetable	Average number of head shots (color 2)	Shots
33 (21H)	Resetable	Number of carriage operations	Times
34 (22H)	Resetable	Number of pump operations	Times
35 (23H)	Resetable	Number of line feeds (for slip)	Lines
36 (24H)	Resetable	Average number of head shots (color 3)	Shots
37 (25H)	Resetable	Average number of head shots (color 4)	Shots
38 (26H)	Resetable	Number of an Ink Cartridge replacing	Times
39 (27H)	Resetable	Number of manual cleaning	Times
50 (32H)	Resetable	Number of autocutter operations	Times
51 (33H)	Resetable	Number of Release	Times
70 (46H)	Resetable	Printer operation time	Hours
158 (9EH)	Cumulative	Number of line feeds (for roll paper)	Lines
159 (9FH)	Cumulative	Average number of head shots (color 1)	Shots
160 (A0H)	Cumulative	Average number of head shots (color 2)	Shots
161 (A1H)	Cumulative	Number of carriage operations	Times
162 (A2H)	Cumulative	Number of pump operations	Times
163 (A3H)	Cumulative	Number of line feeds (for slip)	Lines
164 (A4H)	Cumulative	Average number of head shots (color 3)	Shots
165 (A5H)	Cumulative	Average number of head shots (color 4)	Shots
166 (A6H)	Cumulative	Number of an Ink Cartridge replacing	Times
167 (A7H)	Cumulative	Number of manual cleaning	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
179 (B3H)	Cumulative	Number of Release	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-H5000II

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)

Macro Definitions	ON/ OFF	Value	Status
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
10 (0AH)	Resetable	Number of line feeds (for slip paper)	Lines
11 (0BH	Resetable	Number of printed characters (for the face of a slip)	Characters
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
60 (3CH)	Resetable	Number of MICR reads	Times
70 (46H)	Resetable	Printer operation time	Hours
138 (8AH)	Cumulative	Number of line feeds (for slip paper)	Lines
139 (8BH)	Cumulative	Number of printed characters (for the face of a slip)	Characters
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
179 (B3H)	Cumulative	Number of Release	Times
188 (BCH)	Cumulative	Number of MICR reads	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-H5200

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)

Macro Definitions	ON/ OFF	Value	Status
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
10 (0AH)	Resetable	Number of line feeds (for slip paper)	Lines
11 (0BH	Resetable	Number of printed characters (for the face of a slip)	Characters
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
60 (3CH)	Resetable	Number of MICR reads	Times
70 (46H)	Resetable	Printer operation time	Hours
138 (8AH)	Cumulative	Number of line feeds (for slip paper)	Lines
139 (8BH)	Cumulative	Number of printed characters (for the face of a slip)	Characters
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
179 (B3H)	Cumulative	Number of Release	Times
188 (BCH)	Cumulative	Number of MICR reads	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-H6000II

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector

Macro Definitions	ON/ OFF	Value	Status
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_VALIDATION_SELECTED	ON	0x04000000	Validation not selected as the active sheet
	OFF	0x00000000	Validation selected as the active sheet
ASB_PRINT_VALIDATION	ON	0x08000000	Validation printing is impossible
	OFF	0x00000000	Validation printing is possible
ASB_VALIDATION_NO_PAPER	ON	0x40000000	Validation sensor: paper not present
	OFF	0x00000000	Validation sensor: paper present
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
10 (0AH)	Resetable	Number of line feeds (for slip paper)	Lines
11 (0BH	Resetable	Number of printed characters (for the face of a slip)	Characters
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
40 (28H)	Resetable	Number of head movements back and forth (for the back of a slip)	Times
41 (29H)	Resetable	Number of printed characters (for the back of a slip)	Characters
50 (32H)	Resetable	Number of autocutter operations	Times
60 (3CH)	Resetable	Number of MICR reads	Times
61 (3DH)	Resetable	Number of scans of an image (for the slip image scanner)	Times
62 (3EH)	Resetable	Amount of paper fed for scanned images (for the slip image scanner)	Steps
63 (3FH)	Resetable	Number of scans of an image (for the card image scanner)	Times
64 (40H)	Resetable	Amount of paper fed for scanned images (for the card image scanner)	Steps
70 (46H)	Resetable	Printer operation time	Hours
138 (8AH)	Cumulative	Number of line feeds (for slip paper)	Lines
139 (8BH)	Cumulative	Number of printed characters (for the face of a slip)	Characters
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
168 (A8H)	Cumulative	Number of head movements back and forth (for the back of a slip)	Times
169 (A9H)	Cumulative	Number of printed characters (for the back of a slip)	Characters
178 (B2H)	Cumulative	Number of autocutter operations	Times
188 (BCH)	Cumulative	Number of MICR reads	Times
189 (BDH)	Cumulative	Number of scans of an image (for the slip image scanner)	Times
190 (BEH)	Cumulative	Amount of paper fed for scanned images (for the slip image scanner)	Steps
191 (BFH)	Cumulative	Number of scans of an image (for the card image scanner)	Times

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
192 (C0H)	Cumulative	Amount of paper fed for scanned images (for the card image scanner)	Steps
198 (C6H)	Cumulative	Printer operation time	Hours

TM-H6000III

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector

Macro Definitions	ON/ OFF	Value	Status
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_VALIDATION_SELECTED	ON	0x04000000	Validation not selected as the active sheet
	OFF	0x00000000	Validation selected as the active sheet
ASB_PRINT_VALIDATION	ON	0x08000000	Validation printing is impossible
	OFF	0x00000000	Validation printing is possible
ASB_VALIDATION_NO_PAPER	ON	0x40000000	Validation sensor: paper not present
	OFF	0x00000000	Validation sensor: paper present
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
10 (0AH)	Resetable	Number of line feeds (for slip paper)	Lines
11 (0BH	Resetable	Number of printed characters (for the face of a slip)	Characters
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
40 (28H)	Resetable	Number of head movements back and forth (for the back of a slip)	Times
41 (29H)	Resetable	Number of printed characters (for the back of a slip)	Characters
50 (32H)	Resetable	Number of autocutter operations	Times
60 (3CH)	Resetable	Number of MICR reads	Times
61 (3DH)	Resetable	Number of scans of an image (for the slip image scanner)	Times
62 (3EH)	Resetable	Amount of paper fed for scanned images (for the slip image scanner)	Steps
63 (3FH)	Resetable	Number of scans of an image (for the card image scanner)	Times
64 (40H)	Resetable	Amount of paper fed for scanned images (for the card image scanner)	Steps
70 (46H)	Resetable	Printer operation time	Hours
138 (8AH)	Cumulative	Number of line feeds (for slip paper)	Lines
139 (8BH)	Cumulative	Number of printed characters (for the face of a slip)	Characters
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
168 (A8H)	Cumulative	Number of head movements back and forth (for the back of a slip)	Times
169 (A9H)	Cumulative	Number of printed characters (for the back of a slip)	Characters
178 (B2H)	Cumulative	Number of autocutter operations	Times
188 (BCH)	Cumulative	Number of MICR reads	Times
189 (BDH)	Cumulative	Number of scans of an image (for the slip image scanner)	Times
190 (BEH)	Cumulative	Amount of paper fed for scanned images (for the slip image scanner)	Steps
191 (BFH)	Cumulative	Number of scans of an image (for the card image scanner)	Times

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
192 (C0H)	Cumulative	Amount of paper fed for scanned images (for the card image scanner)	Steps
198 (C6H)	Cumulative	Printer operation time	Hours

TM-J2100

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Ink Status

Macro Definitions	ON/ OFF	Value	Ink status
INK_ASB_NEAR_END	ON	0x0001	Remaining ink level is low (color no. 1)
	OFF	0x0000	-
INK_ASB_END	ON	0x0002	Exchange the ink cartridge (color no. 1)
	OFF	0x0000	-
INK_ASB_NO_CARTRIDGE	ON	0x0004	No ink cartridge (color no. 1)
	OFF	0x0000	Ink cartridge present (color no. 1)
INK_ASB_NO_CARTRIDGE2	ON	0x0008	No ink cartridge (color no. 2)
	OFF	0x0000	Ink cartridge present (color no. 2)
INK_ASB_CLEANING	ON	0x0020	Cleaning
	OFF	0x0000	-
INK_ASB_NEAR_END2	ON	0x0100	Remaining ink level is low (color no. 2)
	OFF	0x0000	-
INK_ASB_END2	ON	0x0200	Exchange the ink cartridge (color no. 2)
	OFF	0x0000	-

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	Overcurrent error generated
	OFF	0x00	Overcurrent error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Internal circuit connection error (thermostat)
	OFF	0x00	No internal circuit connection error (thermostat)
1	ON	0x02	Internal circuit connection error (head high voltage)
	OFF	0x00	No internal circuit connection error (head high voltage)
2	ON	0x04	Internal circuit connection error (head low voltage)
	OFF	0x00	No internal circuit connection error (head low voltage)
3	ON	0x08	Internal circuit connection error (EEPROM)
	OFF	0x00	No internal circuit connection error (EEPROM)
4	ON	0x10	Carriage drive count error
	OFF	0x00	No carriage drive count error
5	ON	0x20	Internal circuit connection error (number of pump driving)
	OFF	0x00	No internal circuit connection error (number of pump driving)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

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 (when (Msw8-8) is on)
	OFF	0x00	No roll paper cover open error (when (Msw8-8) is on)
2	-	0x00	Reserved (fixed to 0)
3	-	0x00	Reserved (fixed to 0)
4	ON	0x10	Carriage home position detection error
	OFF	0x00	No carriage home position detection error
5	ON	0x20	Unsynchronized carriage detection error
	OFF	0x00	No unsynchronized carriage detection error
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

Bit	OFF	Value	Status
0	ON	0x01	Roll paper cover open error (when (Msw8-8) is off)
	OFF	0x00	No roll paper cover open error (when (Msw8-8) is off)
1	ON	0x02	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	ON	0x04	Low head temperature error generated
	OFF	0x00	Low head temperature error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
30 (1EH)	Resetable	Number of line feeds (for roll paper)	Lines
31 (1FH)	Resetable	Average number of head shots (color 1)	Shots
32 (20H)	Resetable	Average number of head shots (color 2)	Shots
33 (21H)	Resetable	Number of carriage operations	Times
34 (22H)	Resetable	Number of pump operations	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
158 (9EH)	Cumulative	Number of line feeds (for roll paper)	Lines
159 (9FH)	Cumulative	Average number of head shots (color 1)	Shots
160 (A0H)	Cumulative	Average number of head shots (color 2)	Shots
161 (A1H)	Cumulative	Number of carriage operations	Times
162 (A2H)	Cumulative	Number of pump operations	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-J7100

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)

Macro Definitions	ON/ OFF	Value	Status
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Ink Status

Macro Definitions	ON/ OFF	Value	Ink status
INK_ASB_NEAR_END	ON	0x0001	Remaining ink level is low (color no. 1)
	OFF	0x0000	-
INK_ASB_END	ON	0x0002	Exchange the ink cartridge (color no. 1)
	OFF	0x0000	-
INK_ASB_NO_CARTRIDGE	ON	0x0004	No ink cartridge (color no. 1)
	OFF	0x0000	Ink cartridge present (color no. 1)
INK_ASB_NO_CARTRIDGE2	ON	0x0008	No ink cartridge (color no. 2)
	OFF	0x0000	Ink cartridge present (color no. 2)
INK_ASB_CLEANING	ON	0x0020	Cleaning
	OFF	0x0000	-
INK_ASB_NEAR_END2	ON	0x0100	Remaining ink level is low (color no. 2)
	OFF	0x0000	-
INK_ASB_END2	ON	0x0200	Exchange the ink cartridge (color no. 2)
	OFF	0x0000	-

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	Overcurrent error generated
	OFF	0x00	Overcurrent error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Internal circuit connection error (thermostat)
	OFF	0x00	No internal circuit connection error (thermostat)
1	ON	0x02	Internal circuit connection error (head high voltage)
	OFF	0x00	No internal circuit connection error (head high voltage)
2	ON	0x04	Internal circuit connection error (head low voltage)
	OFF	0x00	No internal circuit connection error (head low voltage)
3	ON	0x08	Internal circuit connection error (EEPROM)
	OFF	0x00	No internal circuit connection error (EEPROM)
4	ON	0x10	Carriage drive count error
	OFF	0x00	No carriage drive count error
5	ON	0x20	Internal circuit connection error (number of pump driving)
	OFF	0x00	No internal circuit connection error (number of pump driving)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

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 (when (Msw8-8) is on)
	OFF	0x00	No roll paper cover open error (when (Msw8-8) is on)
2	-	0x00	Reserved (fixed to 0)
3	-	0x00	Reserved (fixed to 0)
4	ON	0x10	Carriage home position detection error
	OFF	0x00	No carriage home position detection error
5	ON	0x20	Unsynchronized carriage detection error
	OFF	0x00	No unsynchronized carriage detection error
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

Bit	OFF	Value	Status
0	ON	0x01	Roll paper cover open error (when (Msw8-8) is off)
	OFF	0x00	No roll paper cover open error (when (Msw8-8) is off)
1	ON	0x02	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	ON	0x04	Low head temperature error generated
	OFF	0x00	Low head temperature error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
30 (1EH)	Resetable	Number of line feeds (for roll paper)	Lines
31 (1FH)	Resetable	Average number of head shots (color 1)	Shots
32 (20H)	Resetable	Average number of head shots (color 2)	Shots
33 (21H)	Resetable	Number of carriage operations	Times
34 (22H)	Resetable	Number of pump operations	Times
35 (23H)	Resetable	Number of line feeds (for slip)	Lines
50 (32H)	Resetable	Number of autocutter operations	Times
60 (3CH)	Resetable	Number of MICR reads	Times
61 (3DH)	Resetable	Number of endorsement printing	Times
70 (46H)	Resetable	Printer operation time	Hours
158 (9EH)	Cumulative	Number of line feeds (for roll paper)	Lines
159 (9FH)	Cumulative	Average number of head shots (color 1)	Shots
160 (A0H)	Cumulative	Average number of head shots (color 2)	Shots
161 (A1H)	Cumulative	Number of carriage operations	Times
162 (A2H)	Cumulative	Number of pump operations	Times
163 (A3H)	Cumulative	Number of line feeds (for slip)	Lines
178 (B2H)	Cumulative	Number of autocutter operations	Times
188 (BCH)	Cumulative	Number of MICR reads	Times
189 (BDH)	Cumulative	Number of endorsement printing	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-J7500/7600

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)

Macro Definitions	ON/ OFF	Value	Status
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Ink Status

Macro Definitions	ON/ OFF	Value	Ink status
INK_ASB_NEAR_END	ON	0x0001	Remaining ink level is low (color no. 1)
	OFF	0x0000	-
INK_ASB_END	ON	0x0002	Exchange the ink cartridge (color no. 1)
	OFF	0x0000	-
INK_ASB_NO_CARTRIDGE	ON	0x0004	No ink cartridge (color no. 1)
	OFF	0x0000	Ink cartridge present (color no. 1)
INK_ASB_NO_CARTRIDGE2	ON	0x0008	No ink cartridge (color no. 2)
	OFF	0x0000	Ink cartridge present (color no. 2)
INK_ASB_CLEANING	ON	0x0020	Cleaning
	OFF	0x0000	-
INK_ASB_NEAR_END2	ON	0x0100	Remaining ink level is low (color no. 2)
	OFF	0x0000	-
INK_ASB_END2	ON	0x0200	Exchange the ink cartridge (color no. 2)
	OFF	0x0000	-

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	-	0x00	Reserved
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Internal circuit connection error (thermostat)
	OFF	0x00	No internal circuit connection error (thermostat)
1	ON	0x02	Internal circuit connection error (head high voltage)
	OFF	0x00	No internal circuit connection error (head high voltage)
2	ON	0x04	Internal circuit connection error (head low voltage)
	OFF	0x00	No internal circuit connection error (head low voltage)
3	ON	0x08	Internal circuit connection error (EEPROM)
	OFF	0x00	No internal circuit connection error (EEPROM)
4	-	0x00	Reserved
5	ON	0x20	Internal circuit connection error (number of pump driving)
	OFF	0x00	No internal circuit connection error (number of pump driving)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

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 (when (Msw8-8) is on)
	OFF	0x00	No roll paper cover open error (when (Msw8-8) is on)
2	ON	0x04	Cut sheet ejection error
	OFF	0x00	No cut sheet ejection error
3	-	0x00	Reserved (fixed to 0)
4	ON	0x10	Carriage home position detection error
	OFF	0x00	No carriage home position detection error
5	ON	0x20	Unsynchronized carriage detection error
	OFF	0x00	No unsynchronized carriage detection error
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

Bit	OFF	Value	Status
0	ON	0x01	Roll paper cover open error (when (Msw8-8) is off)
	OFF	0x00	No roll paper cover open error (when (Msw8-8) is off)
1	ON	0x02	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	ON	0x04	Low head temperature error generated
	OFF	0x00	Low head temperature error not generated
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
30 (1EH)	Resetable	Number of line feeds (for roll paper)	Lines
31 (1FH)	Resetable	Average number of head shots (color 1)	Shots
32 (20H)	Resetable	Average number of head shots (color 2)	Shots
33 (21H)	Resetable	Number of carriage operations	Times
34 (22H)	Resetable	Number of pump operations	Times
35 (23H)	Resetable	Number of line feeds (for slip)	Lines
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
158 (9EH)	Cumulative	Number of line feeds (for roll paper)	Lines
159 (9FH)	Cumulative	Average number of head shots (color 1)	Shots
160 (A0H)	Cumulative	Average number of head shots (color 2)	Shots
161 (A1H)	Cumulative	Number of carriage operations	Times
162 (A2H)	Cumulative	Number of pump operations	Times
163 (A3H)	Cumulative	Number of line feeds (for slip)	Lines
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-L60II

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_LABEL_ERR	ON	0x00000400	A label detection error has occurred
	OFF	0x00000000	A label detection error has not occurred
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-L90

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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

Macro Definitions	ON/ OFF	Value	Status
ASB_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	Overcurrent error generated
	OFF	0x00	Overcurrent error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Thermistor error generated
	OFF	0x00	Thermistor error not generated
1	-	0x00	Reserved
2	-	0x00	Reserved
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

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 recoverable error)
	OFF	0x00	Roll paper cover open error not generated (when specifying recoverable error)
2	ON	0x04	Paper layout error generated
	OFF	0x00	Paper layout error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

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	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	-	0x00	Reserved (fixed to 0)
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-L90Peeler

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_RECOVER_ERR	ON	0x00000400	Recoverable error has occurred
	OFF	0x00000000	Recoverable error has not occurred
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_WAIT_REMOVE_LABEL	ON	0x01000000	Waiting to remove label
	OFF	0x00000000	-

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_LABEL	ON	0x04000000	No paper in the label peeling detector
	OFF	0x00000000	Paper is in the label peeling detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	Overcurrent error generated
	OFF	0x00	Overcurrent error not generated
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Thermistor error generated
	OFF	0x00	Thermistor error not generated
1	-	0x00	Reserved
2	-	0x00	Reserved
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

Bit	ON/ OFF	Value	Status
0	-	0x00	Reserved
1	ON	0x02	Roll paper cover open error generated (when specifying recoverable error)
	OFF	0x00	Roll paper cover open error not generated (when specifying recoverable error)
2	ON	0x04	Paper layout error generated
	OFF	0x00	Paper layout error not generated
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

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	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	-	0x00	Reserved
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-P60

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_BATTERY_OFFLINE	ON	0x00000004	Offline due to a weak battery
	OFF	0x00000000	-
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
ASB_PAPER_END	ON	0x00040000	Roll paper is not detected
	OFF	0x00000000	Roll paper is detected
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-P60 Peeler

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_BATTERY_OFFLINE	ON	0x00000004	Offline due to a weak battery
	OFF	0x00000000	-
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
ASB_PAPER_END	ON	0x00040000	Roll paper is not detected
	OFF	0x00000000	Roll paper is detected
ASB_WAIT_REMOVE_LABEL	ON	0x01000000	Waiting to remove label
	OFF	0x00000000	-
ASB_NO_LABEL	ON	0x04000000	No paper in the label peeling detector
	OFF	0x00000000	Paper is in the label peeling detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-T70

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-T88III

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-T88IV

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-T88IV ReStick

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-T90

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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

Macro Definitions	ON/ OFF	Value	Status
ASB_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Offline Cause

First Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	CPU execution error generated
	OFF	0x00	CPU execution error not generated
1	ON	0x02	ROM error generated
	OFF	0x00	ROM error not generated
2	ON	0x04	G/A error generated
	OFF	0x00	G/A error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (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	Overcurrent error generated
	OFF	0x00	Overcurrent error not generated
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Third Byte

Bit	ON/ OFF	Value	Status
0	ON	0x01	Thermistor error generated
	OFF	0x00	Thermistor error not generated
1	-	0x00	Reserved
2	-	0x00	Reserved
3	-	0x00	Reserved
4	-	0x00	Reserved
5	-	0x00	Reserved
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fourth Byte

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 recoverable error)
	OFF	0x00	Roll paper cover open error not generated (when specifying recoverable error)
2	-	0x00	Reserved (fixed to 0)
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Fifth Byte

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	High head temperature error generated
	OFF	0x00	High head temperature error not generated
2	-	0x00	Reserved (fixed to 0)
3	-	0x00	Reserved (fixed to 0)
4	-	0x00	Reserved (fixed to 0)
5	-	0x00	Reserved (fixed to 0)
6	-	0x40	Fixed (fixed to 1)
7	-	0x00	Fixed (fixed to 0)

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
20 (14H)	Resetable	Number of line feeds (for roll paper)	Lines
21 (15H)	Resetable	Number of times head is energized (for roll paper)	Times
50 (32H)	Resetable	Number of autocutter operations	Times
70 (46H)	Resetable	Printer operation time	Hours
148 (94H)	Cumulative	Number of line feeds (for roll paper)	Lines
149 (95H)	Cumulative	Number of times head is energized (for roll paper)	Times
178 (B2H)	Cumulative	Number of autocutter operations	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-U120

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_RECOVER_ERR	ON	0x00000400	Recoverable error has occurred
	OFF	0x00000000	Recoverable error has not occurred
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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

Macro Definitions	ON/ OFF	Value	Status
ASB_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U120II

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_RECOVER_ERR	ON	0x00000400	Recoverable error has occurred
	OFF	0x00000000	Recoverable error has not occurred
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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

Macro Definitions	ON/ OFF	Value	Status
ASB_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U220

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U230

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U295

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
ASB_OFF_LINE	ON	0x00000008	Offline status
	OFF	0x00000000	Online status
ASB_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_SLIP_BOF_2	ON	0x00200000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_TOF_2	ON	0x00400000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U375

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_HEAD_TEMPERATURE_ERR	ON	0x00004000	High head temperature error has occurred
	OFF	0x00000000	High head temperature error has not occurred
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_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_VALIDATION_SELECTED	ON	0x04000000	Validation not selected as the active sheet
	OFF	0x00000000	Validation selected as the active sheet

Macro Definitions	ON/ OFF	Value	Status
ASB_PRINT_VALIDATION	ON	0x08000000	Validation printing is impossible
	OFF	0x00000000	Validation printing is possible
ASB_VALIDATION_BOF	ON	0x40000000	No paper in the validationBOF
	OFF	0x00000000	Paper in the validation BOF
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U590

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

TM-U675

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_WAIT_ON_LINE	ON	0x00000100	Waiting for online recovery
	OFF	0x00000000	Not waiting for online recovery
ASB_PANEL_SWITCH	ON	0x00000200	Panel switch is ON
	OFF	0x00000000	Panel switch is OFF
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_AUTORECOVER_ERR	ON	0x00004000	Auto recovery error generated
	OFF	0x00000000	Auto recovery error not generated
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector

Macro Definitions	ON/ OFF	Value	Status
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_VALIDATION_SELECTED	ON	0x04000000	Validation not selected as the active sheet
	OFF	0x00000000	Validation selected as the active sheet
ASB_PRINT_VALIDATION	ON	0x08000000	Validation printing is impossible
	OFF	0x00000000	Validation printing is possible
ASB_VALIDATION_TOF	ON	0x20000000	No paper in the validation TOF
	OFF	0x00000000	Paper in the validation TOF
ASB_VALIDATION_BOF	ON	0x40000000	No paper in the validation BOF
	OFF	0x00000000	Paper in the validation BOF
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler

Maintenance Counter

Counter Number (readno)	Resetability	Counter (readcounter)	Unit
10 (0AH)	Resetable	Number of line feeds (for slip paper)	Lines
11 (0BH	Resetable	Number of printed characters (for the face of a slip)	Characters
50 (32H)	Resetable	Number of autocutter operations	Times
60 (3CH)	Resetable	Number of MICR reads	Times
70 (46H)	Resetable	Printer operation time	Hours
138 (8AH)	Cumulative	Number of line feeds (for slip paper)	Lines
139 (8BH)	Cumulative	Number of printed characters (for the face of a slip)	Characters
178 (B2H)	Cumulative	Number of autocutter operations	Times
188 (BCH)	Cumulative	Number of MICR reads	Times
198 (C6H)	Cumulative	Printer operation time	Hours

TM-U950

ASB Status

Macro Definitions	ON/ OFF	Value	Status
ASB_NO_RESPONSE	ON	0x00000001	No printer response
	OFF	0x00000000	Printer response
ASB_PRINT_SUCCESS	ON	0x00000002	Print complete
	OFF	0x00000000	-
ASB_DRAWER_KICK	ON	0x00000004	Status of the drawer kick number 3 connector pin = "H"
	OFF	0x00000000	Status of the drawer kick number 3 connector pin = "L"
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_PAPER_FEED	ON	0x00000040	Paper feed switch is feeding paper
	OFF	0x00000000	Paper feed switch is not feeding paper
ASB_MECHANICAL_ERR	ON	0x00000400	Mechanical error generated
	OFF	0x00000000	Mechanical error not generated
ASB_AUTOCUTTER_ERR	ON	0x00000800	Auto cutter error generated
	OFF	0x00000000	Auto cutter error not generated
ASB_UNREOVER_ERR	ON	0x00002000	Unrecoverable error generated
	OFF	0x00000000	Unrecoverable error not generated
ASB_HEAD_TEMPERATURE_ERR	ON	0x00004000	High head temperature error has occurred
	OFF	0x00000000	High head temperature error has not occurred
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_RECEIPT_END	ON	0x00080000	No paper in the roll paper end detector
	OFF	0x00000000	Paper in the roll paper end detector
ASB_SLIP_TOF	ON	0x00200000	No paper in TOF detector
	OFF	0x00000000	Paper in TOF detector
ASB_SLIP_BOF	ON	0x00400000	No paper in BOF detector
	OFF	0x00000000	Paper in BOF detector

Macro Definitions	ON/ OFF	Value	Status
ASB_SLIP_SELECTED	ON	0x01000000	Slip unselected
	OFF	0x00000000	Slip selected: (active sheet = slip)
ASB_PRINT_SLIP	ON	0x02000000	Slip printing is impossible
	OFF	0x00000000	Slip printing is possible
ASB_SPOOLER_IS_STOPPED	ON	0x80000000	Stop the spooler
	OFF	0x00000000	Operation the spooler