EPSON

TM-S9000II-NW/TM-S2000II-NW Web API Reference Guide

M00152202

Rev.C

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 been 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

Microsoft®, Windows®, Visual Studio® and Internet Explorer® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Android™ and Google Chrome™ are trademarks of Google LLC.

Safari™ is a registered trademark of Apple Inc. in the US and other countries.

Mozilla® and Firefox® are trademarks or registered trademarks of the Mozilla Foundation in the US and other countries.

AirPlay, iPad, iPhone, iPod, iPod classic, iPod nano, iPod touch, and Retina are trademarks of Apple Inc., registered in the U.S. and other countries. iPad Air, iPad mini, and Lightning are trademarks of Apple Inc. The trademark “iPhone” is used with a license from Aiphone K.K.

Apple®, Mac OS®, iTunes® and Xcode® are trademarks of Apple Inc. registered in the US and other countries.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Flex™ is a registered trademark or trademark of Adobe Systems Incorporated in the United States and other countries.

Java™ is a registered trademark of Oracle Corporation, its subsidiaries, and affiliates in the U.S. and other countries.

Eclipse® is a trademark or registered trademark of Eclipse Foundation, Inc.

FileMaker is a trademark of FileMaker, Inc., registered in the U.S.and other countries.

QR Code® is a registered trademark of DENSO WAVE INCORPORATED in Japan and other countries.

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

ESC/POS Command System

Epson has embarked on a global initiative to develop ESC/POS, a unique POS printer command system. ESC/POS contains a wealth of unique commands, many of which are patent-protected. Our system enables the configuration of versatile POS systems with a high level of scalability. In addition to being compatible with most Epson POS printers and displays, the flexibility provided by this unique control system facilitates ease of future upgrades. This functionality and convenience of use are appreciated 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.

NOTE	Indicates supplementary explanations and information you should know.

About this Manual

Aim of the Manual

This manual is aimed to provide developers with information required for developing and designing applications using TM-S9000II-NW/TM-S2000II-NW Web API.

Manual Content

The manual is made up of the following sections:

Contents

Overview of Web API and Scan Web API

Overview of Web API

The TM-S9000II-NW/TM-S2000II-NW Web APIs use web applications to control TM-S9000II-NW/TM-S2000II-NW. They consist of the following APIs.

- Scan Web API:
  Controls the check scanner, card scanner, and cut-sheet printing for TM-S9000II-NW/TM-S2000II-NW.
- ePOS-Print XML:
  Controls the receipt printer for TM-S9000II-NW.

Provided Files

File name	Description
TM-S9000IINW_TMS2000IINW_WebAPI_Referenc-Guide_en_revXX.zip	Web API Reference Guide package file.
TM-S9000II_2000II_webapi_rg_en_revA.pdf	This Document.
ePOS-Print_Sample_XML_V1.0.0E.zip	ePOS-Print operation check tool.
EpsonScanWebAPI.v1.0.0.postman_collection.json	Configuration file for Postman.
scan-webapi-sample_v1.0.0.zip	Scan Web API sample program.

Enabling Encryption (HTTPS)

Check the followings for HTTPS communication.

- Prepare a certificate and secret key for the server, then register them to the printer.
- Common name of the server certificate and the host name of the printer must be the same. Otherwise, HTTPS communication will fail with an error.
- Embed a CA (certificate authority) certificate, which is used to authenticate the CA signature on the server certificate, in your program. Otherwise, HTTPS communication will fail with an error.
- When creating an iOS application, make sure to satisfy the requirements of ATS (App Transport Security).
- Comply with requirements for HTTPS communication of your application development environment.
- To send print data to the printer from an application on a HTTPS Web site, make sure to use HTTPS instead of HTTP to communicate with the printer. Otherwise, the communication will fail with an error.

Overview of Scan Web API

Scan Web API is a generic term that refers to using your own application for check scanning, image capture, and endorsement printing without drivers.

With the use of Scan Web API, you can securely control Epson check scanners from any client terminal with a network connection.

Functions

Scan Web API provides the following functions for client terminals with a network connection.

❏ Scanning

- Scan Check
- Scan Photo ID
- Image porcessing (Compress, binarize, skew etc)
- Read MICR
- Endorsement (Physical/Electronic)
- Ring buzzer
- IQA processing
- Read OCR A/B
- Read 1D barcode

❏ Printing

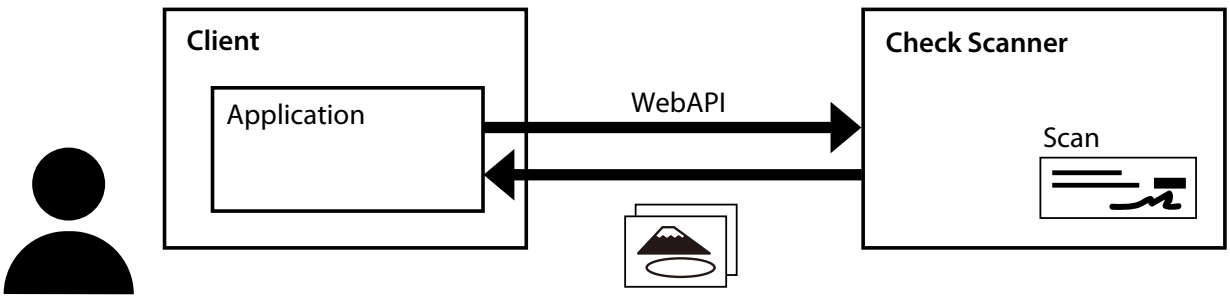
- Print cut sheet

❏ Maintenance

- Reset device
- Cleaning MICR
- Cleaning Ink head
- Get/Reset maintenance counter
- Get log

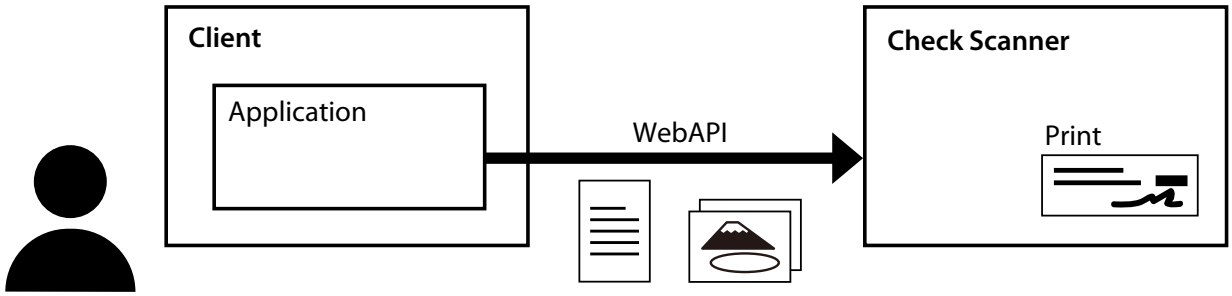
Overview of Scanning Process

The following diagram shows how to use Scan Web API and a scanner to scan and capture a check as image data.



Overview of Printing Process

The following diagram shows how to use Scan Web API for cut-sheet printing.
Unlike scanning, image data is not captured with cut-sheet printing.



Explanation of Terms

Term	Description
Document	Check or photo ID card to be scanned
Transaction Number	Number assigned to a scan target

API list

API Name	Methond	URI	Page
Connect	POST	/api/connect	p.23
KeepAlive	POST	/api/keepalive	p.25
Disconnect	POST	/api/disconnect	p.27
ScanSetting(Check)	POST	/api/scan/setting/check	p.29
ScanSetting(Card)	POST	/api/scan/setting/card	p.44
ScanStart(Check)	POST	/api/scan/start/check	p.49
ScanStart(Card)	POST	/api/scan/start/card	p.51
ScanCancel	POST	/api/scan/cancel	p.53
GetDocumentList	GET	/api/docs	p.55
DeleteAllDocuments	DELETE	/api/docs	p.63
GetImage	GET	/api/docs/{transaction_number}/{image_file_name}	p.65
PrintCutSheet	POST	/api/print/cut_sheet	p.67
GetPrintStatus	GET	/api/print/status	p.71
PrintCancel	POST	/api/print/cancel	p.73
GetDeviceStatus	GET	/api/device/status	p.75
DeviceReset	POST	/api/device/reset	p.77
GetDeviceLog	GET	/api/device/log	p.79
MICRCleaning	POST	/api/device/cleaning/micr	p.80

API Name	Methond	URI	Page
HeadCleaning	POST	/api/device/cleaning/head	p.82
GetCounter	GET	/api/device/counter	p.84
ResetCounter	DELETE	/api/device/counter/{counter_name}	p.87
GetDefaultScanSetting(Check)	GET	/api/scan/setting/check	p.89
GetDefaultScanSetting(Card)	GET	/api/scan/setting/card	p.91
SaveDefaultScanSetting(Check)	PUT	/api/scan/setting/check	p.93
SaveDefaultScanSetting(Card)	PUT	/api/scan/setting/card	p.95
ResetDefaultScanSetting(Check)	DELETE	/api/scan/setting/check	p.97
ResetDefaultScanSetting(Card)	DELETE	/api/scan/setting/card	p.99

Operating Environment

Web browser

- Mozilla Firefox 91 or later
- Google Chrome 89 or later
- Microsoft Edge 91 or later
- Safari on iOS 13 or later

* Windows Internet Explorer is not supported.

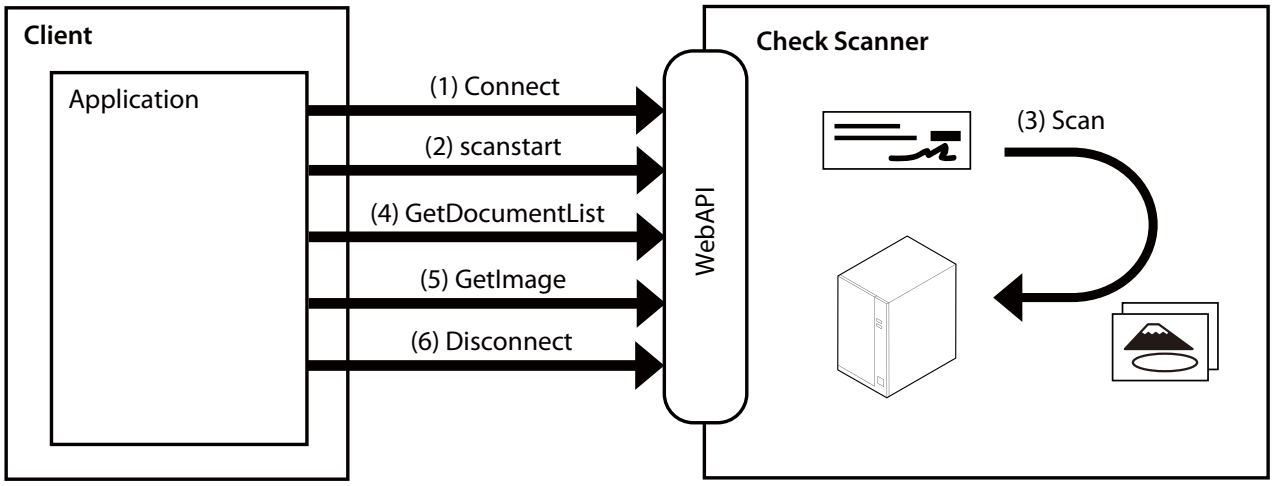
Restrictions

If the device's TCP 9100 port is being used by a print driver or utility software, the scanning function of this Web API cannot be used.

Scan Web API Programming Guide

Scanning Flow

The following figure shows the work flow for using the Scan Web API to scan a check with the client terminal.



If a "(1) Connect" command is received, Scan Web API issues a token. This allows the user to use the scanner exclusively.

If a scan command is received with "(2) ScanStart", the Scan Web API performs continuous scanning of checks.

The processes performed with "(3) Scan" include image processing, MICR reading, endorsement printing, IQA, OCR character reading, and barcode reading.

The scanned image data is retained inside the scanner device, and a list of the document information that is retained is returned by the "(4) GetDocumentList" command.

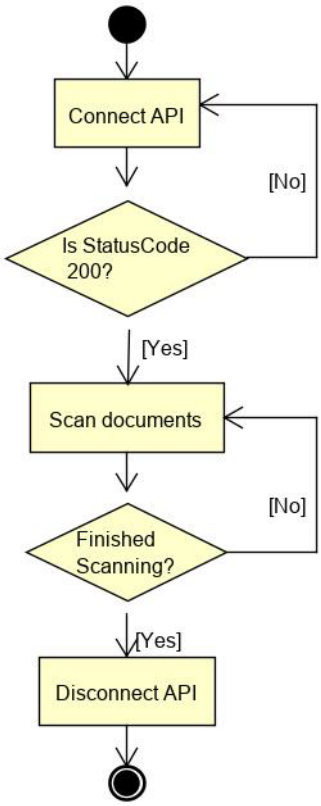
Scanned image data is obtained by calling "(5) GetImage" based on the list of document information.

If a "(6) Disconnect" command is received, the network scanner deletes all document information and image data that has been retained, which allows the device to be shared securely.

Document Scanning Flow

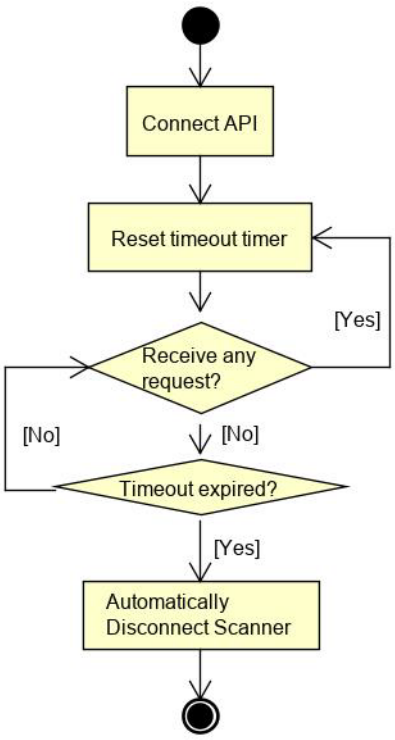
Connecting/Disconnecting

In order to scan a document, a connection must be made from the client to the scanner. Once a connection is made, the client can scan documents. When scanning is complete, the client is disconnected so that the scanner can be used by other clients.



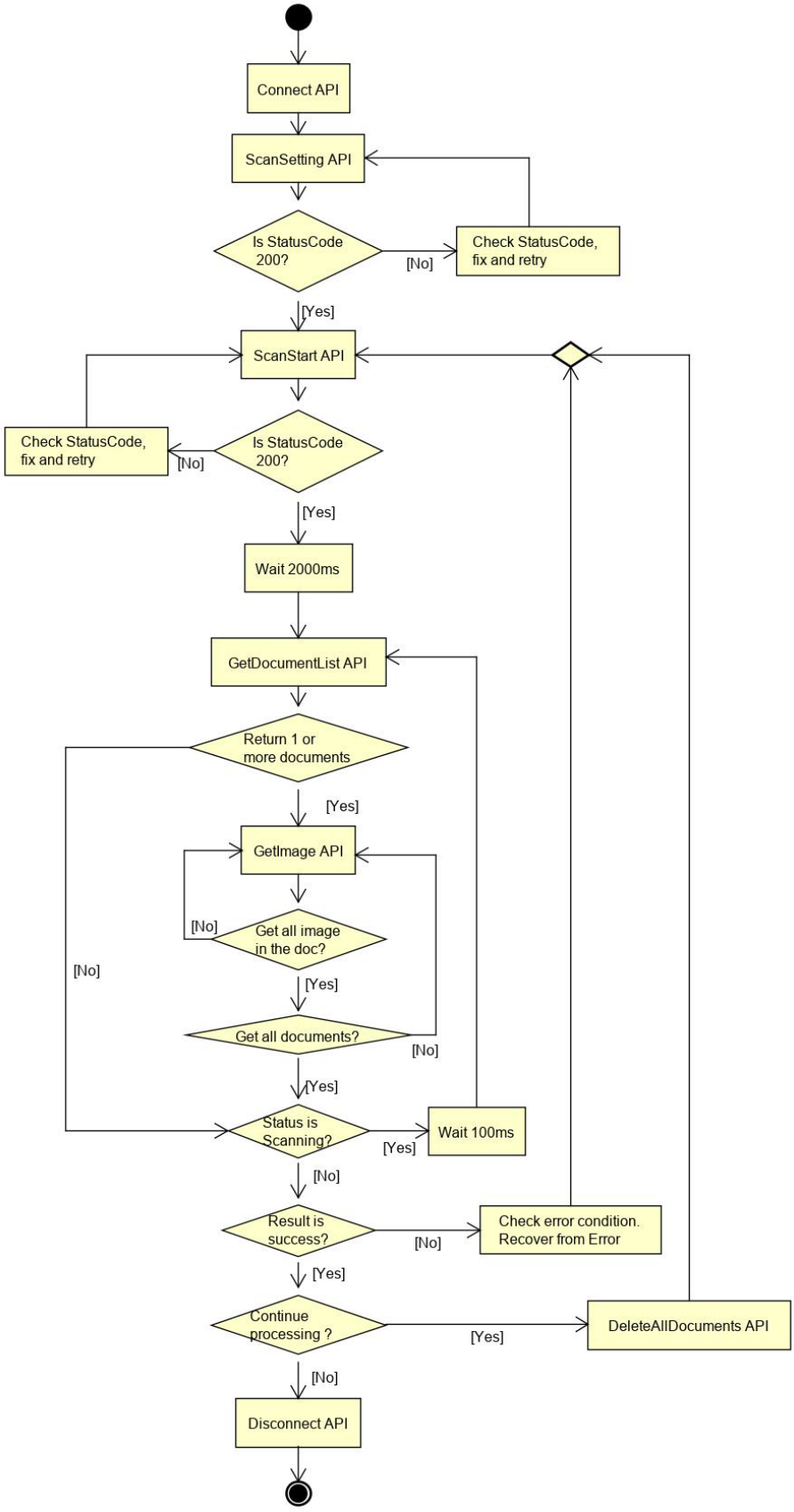
Timeout for Connecting/Disconnecting

If the client does not issue a request to the device for a certain period of time (60 seconds by default), the device will automatically disconnect. This timeout period is reset each time a request is received.



Scanning Documents

The following work flow is used to scan a check.



- Use the GetDocumentList API to obtain the Status and Result.

Deleting Image Data

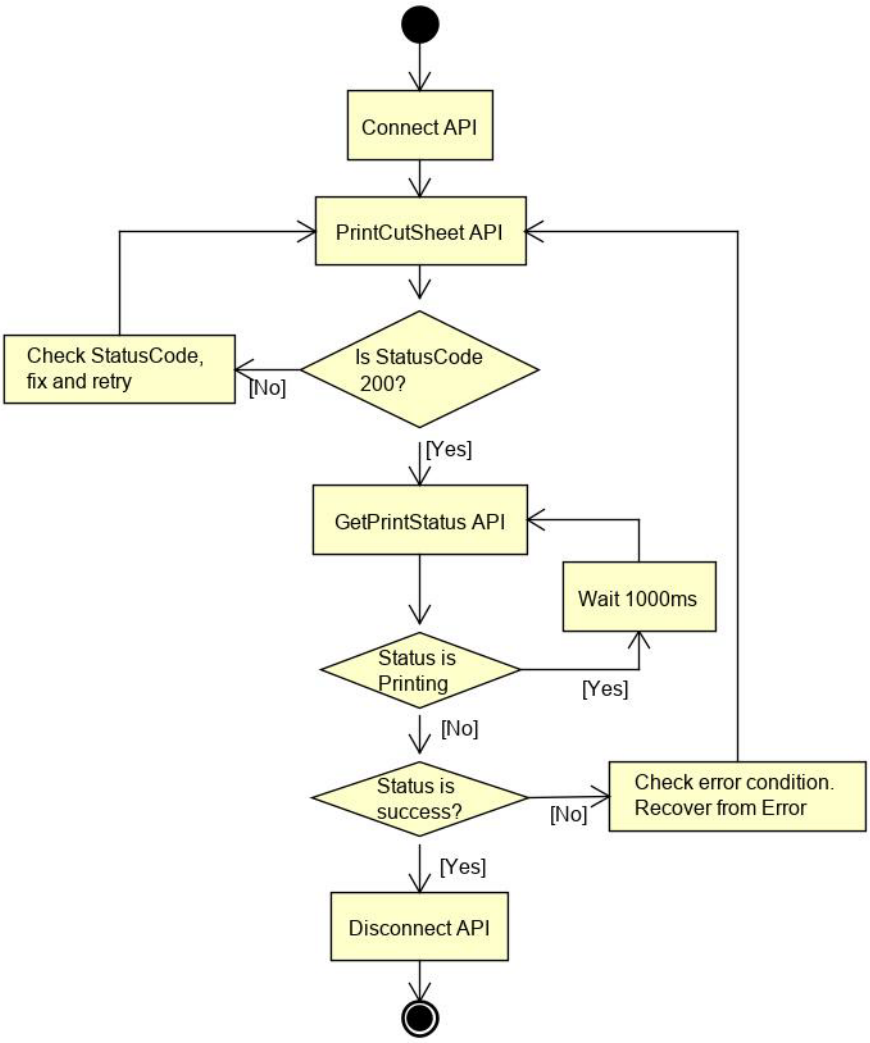
Scanned image data is deleted from the scanner in the following instances.

- If the image data that is retained during the scanning process exceeds the limit, images are deleted in order from oldest to newest until the limit is no longer exceeded. (The limit is 2000 images or 400 MB, whichever comes first.)
- If the DeleteAllDocuments API is run, all image data that is retained will be deleted.
- If the Disconnect API is run, all image data that is retained will be deleted.
- If a connection timeout occurs, all image data that is retained at that time will be deleted.
- If a scan is performed with a transaction number that already exists, the previously scanned document will be deleted.

NOTE	• Image information that has been obtained with the GetImage API is excluded from the GetDocumentList API response, but the image data is not deleted. • When a series of scans is complete, run the DeleteAllDocuments API or Disconnect API to delete the image data.

Printing Flow

The following work flow is used for cut-sheet printing.



ScanWebAPI Reference

Common API Specifications

URI

http(s)://{%IP Address%}/api/{%resource%}

Character code

UTF-8 (BOM not included)

Response

(1) HTTP body (Normal response)

When API processing is completed successfully, an HTTP status code (in the 200s) and a response object in JSON format are returned.

(2) HTTP body (Error response)

If an API processing error occurs, an HTTP status code (in the 400s and 500s) and an error object in JSON format as shown below are returned. Refer to "Common error" on page 18 for more information about common API errors.

If an error response (status code 4xx, 5xx) is returned, an error resource with the following properties is returned.

```
{
   "code" : "..."
}
```

Common error

Status code	Error code string	Description
400	parse_error	The specified resource and query string format are not valid (parsing failed).
400	validation_error	The specified resource and query string value are not appropriate (validation failed).
401	access_token_verification_failed	The access token is disabled.
405	method_not_allowed	The specified HTTP method is not supported or not valid.
415	unsupported_media_type	The specified Content-Type header is not supported or not valid.
500	internal_server_error	An unknown error occurred.

* Other errors are defined in each API.

If a validation_error occurs, the detail property can be used to obtain details as shown in the example below.

```
{
   "code": "validation_error",
   "detail": {
      "key": "/resolution",
      "value": "\"XXXXdpi\"",
      "message": "instance not found in required enum"
   }
}
```

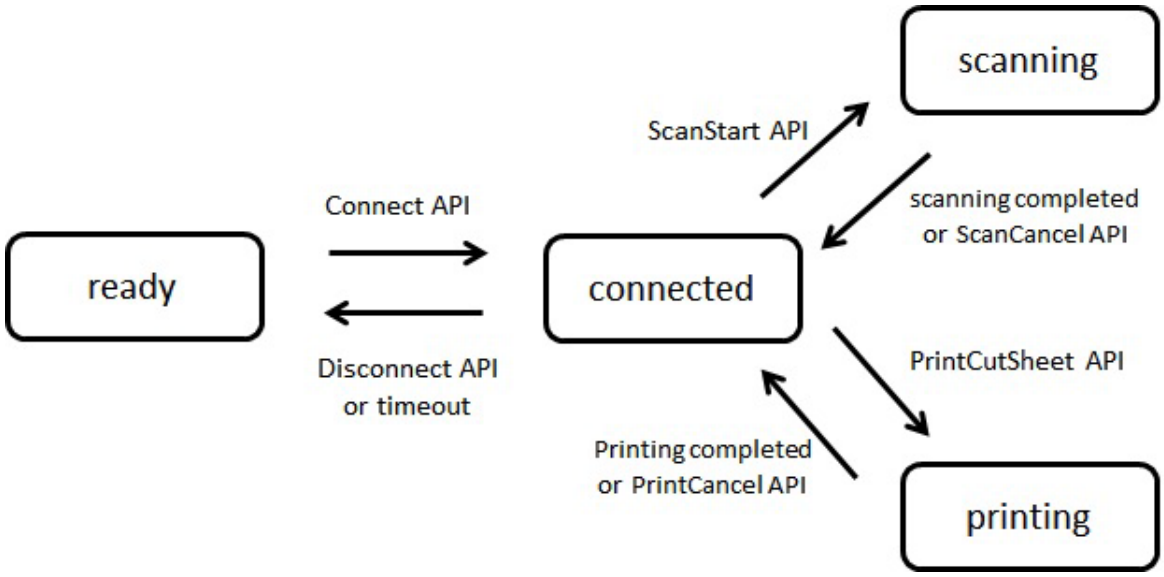
Property	Description
key	Key name where validation error occurred
value	Value where validation error occurred
message	Description of error

State of Scanner

The scanner has the following four states. The state changes when the Web API is called and according to scanning operations.

State	Description
ready	You can obtain exclusive access to the scanner. To start using the scanner, run the connection API to obtain exclusive access.
connected	The scanner is connected. You can run a variety of APIs, such as for scanning and printing. The scanner will return to the ready state if the disconnect API is run or if a connection timeout occurs.
scanning	The scanning process (including endorsement printing) is in progress. When the scanning process is complete, the scanner returns to the connected state.
printing	Cut-sheet printing is in progress. When the printing process is complete, the scanner returns to the connected state.

state change



State Matrix

The following status codes and error code strings are returned when an API is called in each state.

API / state		ready	connected	scanning	printing
Connect	status code	200	400	400	400
	error code string	-	device_busy	device_busy	device_busy
KeepAlive	status code	401	200	200	200
	error code string	access_token_verification_failed	-	-	-
Disconnect	status code	401	200	200*1	200*1
	error code string	access_token_verification_failed	-	-	-

API / state		ready	connected	scanning	printing
ScanSetting(Check)	status code	401	200	400	200
	error code string	access_token_verification_failed	-	device_busy	-
ScanSetting(Card)	status code	401	200	400	200
	error code string	access_token_verification_failed	-	device_busy	-
ScanStart(Check)	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
ScanStart(Card)	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
ScanCancel	status code	401	400	200	400
	error code string	access_token_verification_failed	not_scanning	-	not_scanning
GetDocumentList	status code	401	200	200	200
	error code string	access_token_verification_failed	-	-	-
DeleteAllDocuments	status code	401	200	200	200
	error code string	access_token_verification_failed	-	-	-
GetImage	status code	401	200	200	200
	error code string	access_token_verification_failed	-	-	-
PrintCutSheet	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
GetPrintStatus	status code	401	200	200	200
	error code string	access_token_verification_failed	-	-	-
PrintCancel	status code	401	400	400	200
	error code string	access_token_verification_failed	not_printing	not_printing	-
GetDeviceStatus	status code	200	200	200[*2]	200[*2]
	error code string	-	-	-	-
DeviceReset	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
GetDeviceLog	status code	200	200	200	200
	error code string	-	-	-	-
MICRCleaning	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
HeadCleaning	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy

API / state		ready	connected	scanning	printing
GetCounter	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
ResetCounter	status code	401	200	400	400
	error code string	access_token_verification_failed	-	device_busy	device_busy
GetDefaultScanSetting(Check)	status code	200	200	400	200
	error code string	-	-	device_busy	-
GetDefaultScanSetting(Card)	status code	200	200	400	200
	error code string	-	-	device_busy	-
SaveDefaultScanSetting(Check)	status code	200	200	400	200
	error code string	-	-	device_busy	-
SaveDefaultScanSetting(Card)	status code	200	200	400	200
	error code string	-	-	device_busy	-
ResetDefaultScanSetting(Check)	status code	200	200	400	200
	error code string	-	-	device_busy	-
ResetDefaultScanSetting(Card)	status code	200	200	400	200
	error code string	-	-	device_busy	-

*1 If scanning or printing is in progress, the process is aborted.

*2 The ink status cannot be obtained.

Control from Multiple Clients

When Client 1 connects, it gains exclusive access until the connect timeout is reached or the client disconnects.

For example, if Client 1 is using the scanner, the "connect" command from Client 2 receives "status code: 400, error code string: device_busy" as the response, and all other APIs receive "status code: 401, error code string: access_token_verification_failed" as the response.

Connect

This API is used to start the use of the scanner.

Exclusive access is obtained and an authentication token and device information are returned.

The timeout specified in the parameters is the number of seconds since the last request from the client that the connection will be maintained. If the client sends a request to the scanner after this timeout period has elapsed, an error is returned. All documents scanned by that client will be deleted when the timeout is reached.

Request

Command URI

/api/connect

HTTP Method

POST

HTTP header

Header name	Required	Value
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Item name	Type	Required	Default value	Description
timeout	Numeric	-	60	Specify the timeout for exclusive access from 10 to 3600 (seconds).

HTTP body(Example)

```
{
   "timeout": 60
}
```

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body


```
{
   "token": "550e8400-e29b-41d4-a716-446655440000",
   "hardware": {
      "device_name": "TM-S2000II-NW",
      "manufacture": "EPSON",
      "serial_number": "ABCD12345678",
      "scan_speed": "200DPM",
      "pocket": "1 pocket"
   },
   "software": {
      "scanner_version": "5.02 ESC/POS",
      "main_version": "1.00",
      "interface1_version": "1.00",
      "interface2_version": "1.00",
      "webapi_version": "1.00"
   }
}
```


Item name	Type	Description
token	String	Authentication token (UUID format)
hardware	Object	Hardware information
device_name	String	Model name
manufacture	String	Manufacturer
serial_number	String	Serial number
scan_speed	String	Scanning speed
pocket	String	Pocket specification
software	Object	Software information
scanner_version	String	Scanner version
main_version	String	Main engine version
interface1_version	String	Interface engine 1 version
interface2_version	String	Interface engine 2 version
webapi_version	String	Web API version

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Could not get exclusive	400	device_busy

KeepAlive

This API is used if the client does not call any other APIs for a long time, but wants to stay connected in order to use the scanner.

This API is used to reset the internal timeout counter for sessions in progress.

Request

Command URI

/api/keepalive

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Authentication error	401	access_token_verification_failed

Disconnect

This API is used to end exclusive access to the scanner.
To release control of the scanner, a current, valid authentication token must be used.
All scanned image data is deleted and does not remain in the scanner.
Any scanning or printing operations that are in progress will be stopped.

Request

Command URI

/api/disconnect

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Authentication error	401	access_token_verification_failed

Set Scan Setting (Check)

This API is used to configure the settings for scanning checks.
These settings are retained until Disconnect is performed.
If a parameter is not specified, the default value is used.

Request

Command URI

/api/scan/setting/check

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Item name	Type	Required	Default value	Description
face	String		both	Select from the following. • both • front • back
images	Array (Object)			This array contains captured image information objects.
format	String		tiff	Select from the following. • jpeg: JPEG format normal compression data • tiff: TIFF format CCITT (Group 4) compressed data • bitmap: Bitmap format uncompressed data • tiff256: TIFF format uncompressed data • jpeg_high: JPEG format high compression (size priority) data • jpeg_low: JPEG format low compression (quality priority) data • jtiff: TIFF format JPEG compressed data

Item name	Type	Required	Default value	Description
type	String		black-and-white	Select from the following. • grayscale • color • black-and-white If light_source is IR, grayscale or black-and-white can be selected. If light_source is RGB/IR and color is selected, only RGB images can be obtained.
ex_option	String		sharp	Select from the following. • sharp: Sharpening • manual: Applies the value of threshold for Black and White.
threshold	Numeric		0	Specify the threshold when type is black-and-white and ex_option is manual. -128≤threshold≤127
resolution	String		200dpi	Select from the following. • 100dpi • 120dpi • 200dpi • 240dpi • 300dpi
brightness	Numeric		0	-100≤brightness≤100 If 0, brightness is not corrected. Look at the scan results and set the appropriate value based on the data.
contrast	Numeric		0	-100≤contrast≤100 If 0, contrast is not corrected. Look at the scan results and set the appropriate value based on the data.
gamma	Numeric		1.0	Select from the following. • 1.0 • 1.8 • 2.2 If 1.0, gamma is not corrected. Look at the scan results and set the appropriate value based on the data.
double_feed_eject	Boolean		true	Set whether to eject the paper when double-feed is detected. If the paper is not ejected, a recoverable error will occur on the main unit. • true: The check is ejected even if double-feed is detected. • false: If double-feed is detected, the check is stopped in the route and a recoverable error occurs.
double_feed_stop	Boolean		true	Set whether to stop scanning when double-feed is detected. • true: Scanning is stopped when double-feed is detected. • false: Scanning continues even if double-feed is detected.

Item name	Type	Required	Default value	Description
light_source	String		RGB	The light source to be used for scan execution is selected. • RGB: Visible light • IR: Infrared light • RGB/IR: Visible and infrared light If images[n].type includes color, RGB and IR images cannot be obtained simultaneously; only RGB images are obtained.
micr	Object			This is an MICR object.
enabled	Boolean		true	Set whether to perform read by MICR.
error_eject	Boolean		true	Select from the following. • true: The check is ejected even if an error is detected. • false: If an error is detected, the check is stopped in the route and a recoverable error occurs.
error_stop	Boolean		true	Set whether to stop scanning if there is an error in the reading results.
baddata_count	Numeric		255	If unrecognized MICR characters are detected as errors, set the number of unrecognized characters that are allowed. 0 ≤ baddata_count ≤ 255
font	String		E13B	Select from the following. • E13B: Magnetic waveform analysis module version(E13B) • CMC7: Magnetic waveform analysis module version(CMC7)
parsing	Boolean		true	Set whether to parse MICR strings when the font is E13B. This setting is always false for CMC7.
clear_spaces	Boolean		false	Clears any spaces in MICR data.
buzzer	Array (Object)		[]	This array contains buzzer condition objects.
type	String	✓		Select the conditions under which the buzzer sounds. • success: The buzzer sounds if the reading is successful. • error: The buzzer sounds if a read error occurs. • double_feed: The buzzer sounds if double-feed is detected.
frequency	String	✓		Select from the following. • 440hz • 880hz • 4000hz
count	Numeric	✓		Number of times the buzzer sounds. 1 ≤ count ≤ 3
endorse	Object			This is an endorsement object.
type	String		none	Select from the following. • none • physical • physical_payee • electronic
error_stop	Boolean		false	Set whether to stop scanning when a print data error is detected.

Item name	Type	Required	Default value	Description
data	Array (Object)		[]	This array contains endorsement printing information objects.
font_type	String		sans-serif	Select from the following. • serif • sans-serif • monospace
bold	Boolean		false	Set whether to use bold text.
font_size	Numeric		10	Font size. 4≤font_size≤72 If type is image, ignore this setting.
top_left_x	Numeric	✓		X-coordinates for print origin position (units: mm) 0 ≤ top_left_x ≤ 235 If endorse.type is physical or physical_payee, only values up to 214 are valid. (If a value greater than 214 is specified, a value of 214 will be used.)
top_left_y	Numeric	✓		Y-coordinates for print origin position (units: mm) 0 ≤ top_left_y ≤ 109 If endorse.type is physical or physical_payee, only values up to 49 are valid. (If a value greater than 49 is specified, a value of 49 will be used.)
width	Numeric	✓		Print width (units: mm) 1 ≤ width ≤ 236 If endorse.type is physical or physical_payee, only values up to 215 are valid. (If a value greater than 215 is specified, a value of 215 will be used.)
height	Numeric	✓		Print height (units: mm) 1 ≤ height ≤ 110 If endorse.type is physical or physical_payee, only values up to 50 are valid. (If a value greater than 50 is specified, a value of 50 will be used.)
direction	String		left_to_right	Select from the following. • left_to_right: From left to right (normal direction) • bottom_to_up: From top to bottom (Rotate 90° clockwise) • right_to_left: From right to left (upside down) • up_to_bottom: From bottom to top (Rotate 90° counterclockwise)
type	String	✓		Select from the following. • text • image
contents	String	✓		If type is text, specify a print string. If type is image, specify Base64 encoded image data. Min. 1 character If type is text : Max. 1024 characters. If type is image : Max. 10485760 characters. The following image formats are supported. • BMP format (Uncompressed image data only) • JPEG format (Baseline DCT, Progressive) • TIFF format (CCITT Group 3/Group 4 compressed data, uncompressed data only)

Item name	Type	Required	Default value	Description
iqa	Object			This is an IQA configuration object.
enabled	Boolean		false	Set whether to perform IQA. If images[n].type includes color, IQA cannot be executed.
error_eject	Boolean		true	Set whether to eject the paper if not_pass is in the IQA execution results. If the paper is not ejected, a recoverable error will occur on the main unit.
error_stop	Boolean		false	Set whether to stop scanning if not_pass is in the IQA execution results.
format	String		tiff	Select from the following. • jpeg: JPEG format normal compression data • tiff: TIFF format CCITT (Group 4) compressed data • bitmap: Bitmap format uncompressed data • tiff256: TIFF format uncompressed data • jpeg_high: JPEG format high compression (size priority) data • jpeg_low: JPEG format low compression (quality priority) data • jtiff: TIFF format JPEG compressed data
type	String		black-and-white	Select from the following. • grayscale • black-and-white
ex_option	String		sharp	Select from the following. • sharp: Sharpening • manual: Applies the value of threshold for Black and White.
threshold	Numeric		0	Specify the threshold when type is black-and-white and ex_option is manual. -128≤threshold≤127
resolution	String		200dpi	Select from the following. • 100dpi • 120dpi • 200dpi • 240dpi • 300dpi
under_size	Boolean		true	Set whether to determine that the conditions for UndersizeImage have occurred.
over_size	Boolean		true	Set whether to determine that the conditions for OversizeImage have occurred.
min_compressed_image_size	Boolean		true	Set whether to determine that the conditions for MinCompressedImageSize have occurred.
max_compressed_image_size	Boolean		true	Set whether to determine that the conditions for MaxCompressedImageSize have occurred.
front_rear_image_mismatch	Boolean		true	Set whether to determine that the conditions for FrontRearImageMismatch have occurred.
image_too_light	Boolean		true	Set whether to determine that the conditions for ImageTooLight have occurred.

Item name	Type	Required	Default value	Description
image_too_dark	Boolean		true	Set whether to determine that the conditions for ImageTooDark have occurred.
horizontal_streaks_present	Boolean		true	Set whether to determine that the conditions for HorizontalStreaksPresent have occurred.
excessive_spot_noise	Boolean		true	Set whether to determine that the conditions for ExcessiveSpotNoise have occurred.
image_out_of_focus	Boolean		true	Set whether to determine that the conditions for ImageOutOfFocus have occurred.
folded_torn_doc_corners	Boolean		true	Set whether to determine that the conditions for FoldedTornDocCorners have occurred.
folded_torn_doc_edges	Boolean		true	Set whether to determine that the conditions for FoldedTornDocEdges have occurred.
doc_framing_error	Boolean		true	Set whether to determine that the conditions for DocFramingError have occurred.
excessive_doc_skew	Boolean		true	Set whether to determine that the conditions for ExcessiveDocSkew have occurred.
carbon_strip_detection	Boolean		true	Set whether to determine that the conditions for CarbonStripDetection have occurred.
piggy_back	Boolean		true	Set whether to determine that the conditions for Piggyback have occurred.

Item name	Type	Required	Default value	Description
ocr_ab	Object			This is an OCR AB information object.
enabled	Boolean		false	Set whether to perform OCR AB reading.
type	String		ocr_a_alphanum	Select from the following. ocr_a_num: OCR-A font, numeric characters only ocr_b_num: OCR-B font, numeric characters only ocr_a_alpha: OCR-A font, alphabetic characters only ocr_b_alpha: OCR-B font, alphabetic characters only ocr_a_alphanum: OCR-A font, alphanumeric characters ocr_b_alphanum: OCR-B font, alphanumeric characters ocr_a_alphanum_wooh: CR-A font, alphanumeric characters (except the letter O) ocr_b_alphanum_wooh: CR-B font, alphanumeric characters (except the letter O) ocr_a_alphanum_wozero: OCR-A font, alphanumeric characters (except the number 0) ocr_b_alphanum_wozero: OCR-B font, alphanumeric characters (except the number 0) ocr_a_symnum: OCR-A font, numeric characters and symbols (except "+") ocr_b_symnum: OCR-B font, numeric characters and symbols (except "+")
direction	String		left_to_right	Select from the following. • left_to_right • bottom_to_up • right_to_left • up_to_bottom
start_x	Numeric	✓		Specify the starting X coordinates for the area to be recognized for OCR. (Units: mm) 0 ≤ start_x ≤ 254 and start_x < end_x
end_x	Numeric	✓		Specify the ending X coordinates for the area to be recognized for OCR. (Units: mm) 1 ≤ end_x ≤ 255 and start_x < end_x
start_y	Numeric	✓		Specify the starting Y coordinates for the area to be recognized for OCR. (Units: mm) 0 ≤ start_y ≤ 255 and start_y < end_y

Item name	Type	Required	Default value	Description
end_y	Numeric	✓		Specify the ending Y coordinates for the area to be recognized for OCR. (Units: mm) 1 ≤ end_y ≤ 256 and start_y < end_y
space_enabled	Boolean		false	Includes spaces in the results.
barcode	Object			This is a barcode reading information object.
enabled	Boolean		false	Set whether to perform barcode reading.
error_eject	Boolean		true	Set whether to eject the paper if a barcode reading error occurs. If the paper is not ejected, a recoverable error will occur on the main unit.
error_stop	Boolean		false	Set whether to stop scanning if a barcode reading error occurs during continuous reading. If error_eject is true and error_stop is true, the paper is ejected and scanning stops.
face	String		front	Select from the following. • both • front • back
types	Array (String)		["CODE128"]	Select from the following. (You can select more than one.) • CODE128 • CODABAR • CODE39 • ITF • EAN/JAN • UPC-A • UPC-E

HTTP body(Example)

```
{
   "face": "both",
   "images": [
      {
         "format": "jpeg",
         "type": "grayscale"
      },
      {
         "format": "tiff",
         "type": "black-and-white"
      }
   ],
   "resolution": "200dpi",
   "endorse": {
      "type": "physical",
      "error_stop": true,
      "data": [
         {
            "font_type": "sans-serif",
            "bold": true,
            "font_size": 10,
            "top_left_x": 0,
            "top_left_y": 0,
            "width": 50,
            "height": 10,
            "direction": "left_to_right",
            "type": "text",
            "contents": "Hello world!"
         }
      ]
   }
}
```

- In this example, the following image capture and endorsement printing settings are configured.

Function	No.	Scan face	File format	Image type
Image Capture	1	Front side	JPEG	Grayscale
	2	Front side	TIFF	Black-and-White
	3	Back side	JPEG	Grayscale
	4	Back side	TIFF	Black-and-White
Endorsement Printing	5	Printed text: Hello world!		

Default value


```
{
   "face": "both",
   "images": [
      {
         "format": "tiff",
         "type": "black-and-white",
         "ex_option": "sharp",
         "threshold": 0
      }
   ],
   "resolution": "200dpi",
   "brightness": 0,
   "contrast": 0,
   "gamma": 1.0,
   "double_feed_eject": true,
   "double_feed_stop": true,
   "light_source": "RGB",
   "micr": {
      "enabled": true,
      "error_eject": true,
      "error_stop": true,
      "baddata_count": 255,
      "font": "E13B",
      "parsing": true,
      "clear_spaces": false
   },
   "buzzer": [],
   "endorse": {
      "type": "none",
      "error_stop": false,
      "data": []
   },
   "iqa": {
      "enabled": false,
      "error_eject": true,
      "error_stop": false,
      "format": "tiff",
      "type": "black-and-white",
      "ex_option": "sharp",
      "threshold": 0,
      "resolution": "200dpi",
      "under_size": true,
      "over_size": true,
      "min_compressed_image_size": true,
      "max_compressed_image_size": true,
      "front_rear_image_mismatch": true,
      "image_too_light": true,
      "image_too_dark": true,
```

```
        "horizontal_streaks_present": true,
        "excessive_spot_noise": true,
        "image_out_of_focus": true,
        "folded_torn_doc_corners": true,
        "folded_torn_doc_edges": true,
        "doc_framing_error": true,
        "excessive_doc_skew": true,
        "carbon_strip_detection": true,
        "piggy_back": true
    },
    "ocr_ab": {
        "enabled": false,
        "type": "ocr_a_alphanum",
        "direction": "left_to_right",
        "start_x": 0,
        "end_x": 255,
        "start_y": 0,
        "end_y": 256,
        "space_enabled": false
    },
    "barcode": {
        "enabled": false,
        "error_eject": true,
        "error_stop": false,
        "face": "front",
        "types": [
            "CODE128"
        ]
    }
}
```


Request - Description of Endorsement Printing Function

Printing a Payee Endorsement

To print a payee endorsement, specify physical_payee for endorse.type.

❏ The length of the paper is measured and the print start position is moved to the right.

❏ This allows the right side of the print data to be set along the right side of the paper when printing.

❏ This function does not perform right justification for the print data.

❏ When combined with direction:up_to_bottom, the paper can be rotated 90 degrees to print a payee endorsement.

❏ If the size of the print data is larger than the printable area of this product:

- The print start position is moved to the left side of the printable area.
- Print data that exceeds the printable area will not be printed.

Sequence Printing Function

❏ This function specifies the print format for transaction numbers. The format of the keywords enclosed in <> is used when printing.

❏ The parameters in the ScanStart API are used to specify the transaction number.

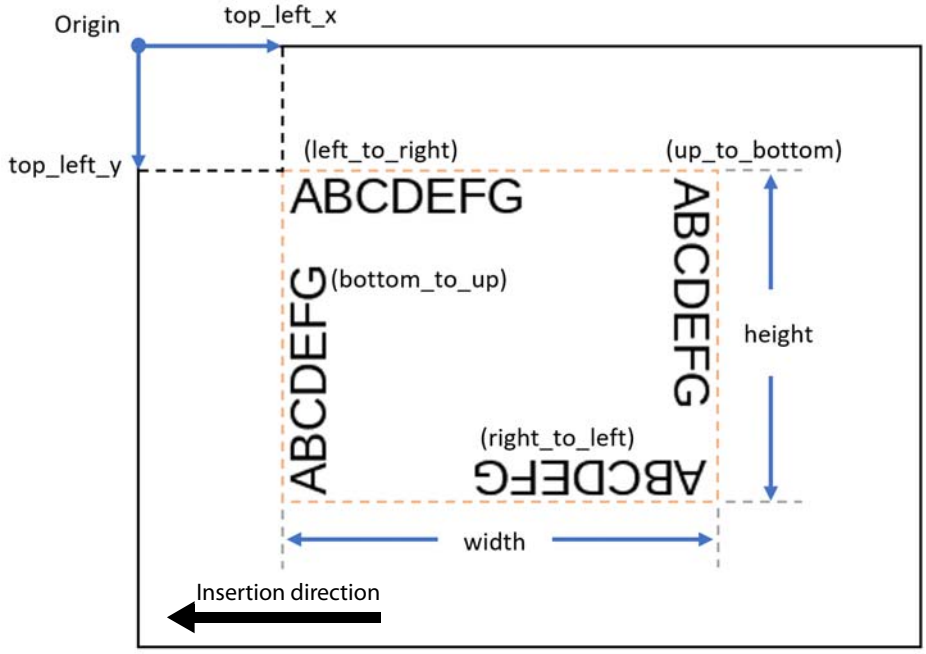
❏ There are three patterns for keywords that can be specified for sequence printing.

Keywords	Description
<0000>	The number of "0"s indicates the number of columns to be printed. The number of columns that can be set is from 1 to 9. If a fewer number of transactions than the number of columns is set, a 0 is automatically added.
<xxxx>	The number of "x"s indicates the number of columns to be printed. The number of columns that can be set is from 1 to 9. If a fewer number of transactions than the number of columns is set, a space is automatically added.
<llll>	(Single-byte lower case "L"): The number of "l"s indicates the number of columns to be printed. The number of columns that can be set is from 1 to 9. If a fewer number of transactions than the number of columns is set, the columns are automatically left justified.

NOTE	• "<" and ">" are used as special symbols. When printing "<" or ">", specify &< or &> respectively. • If a keyword that does not fit the above rules is specified, such as <00xxabc> (a combination of 0s and xs, or characters other than 0 or x), a normal string enclosed in < and > will be output. • For sequential printing, if you specify n digits fewer than the transaction number that is currently set, only the last n digits of the transaction number will actually be printed. <Example> If the current transaction number is 12345 and you specify <xxxx> for sequential printing with 4 columns, "2345" will be printed.

Printing Position and Direction of Endorsement

The following figure shows how an endorsement will appear if the print position and direction are specified.



Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body


```
{
   "face": "both",
   "images": [
      {
         "format": "jpeg",
         "type": "grayscale"
      },
      {
         "format": "tiff",
         "type": "black-and-white"
      }
   ],
   "resolution": "200dpi",
   "brightness": 0,
   "contrast": 0,
   "gamma": 1.0,
   "double_feed_eject": true,
   "double_feed_stop": true,
   "light_source": "RGB",
   "micr": {
      "enabled": true,
      "error_eject": true,
      "error_stop": true,
      "font": "E13B",
      "parsing": true,
      "clear_spaces": false
   },
   "buzzer": [],
   "endorse": {
      "type": "physical",
      "error_stop": true,
      "data": [
         {
            "font_type": "sans-serif",
            "bold": true,
            "font_size": 10,
            "top_left_x": 0,
            "top_left_y": 0,
            "width": 50,
            "height": 10,
            "direction": "left_to_right",
            "type": "text",
            "contents": "Hello world!"
         }
      ]
   },
```

```
    "iqa": {
        "enabled": false,
        "error_eject": true,
        "error_stop": false,
        "format": "tiff",
        "type": "black-and-white",
        "ex_option": "sharp",
        "threshold": 0,
        "resolution": "200dpi",
        "under_size": true,
        "over_size": true,
        "min_compressed_image_size": true,
        "max_compressed_image_size": true,
        "front_rear_image_mismatch": true,
        "image_too_light": true,
        "image_too_dark": true,
        "horizontal_streaks_present": true,
        "excessive_spot_noise": true,
        "image_out_of_focus": true,
        "folded_torn_doc_corners": true,
        "folded_torn_doc_edges": true,
        "doc_framing_error": true,
        "excessive_doc_skew": true,
        "carbon_strip_detection": true,
        "piggy_back": true
    },
    "ocr_ab": {
        "enabled": false,
        "type": "ocr_a_alphanum",
        "direction": "left_to_right",
        "start_x": 0,
        "end_x": 255,
        "start_y": 0,
        "end_y": 256,
        "space_enabled": false
    },
    "barcode": {
        "enabled": false,
        "error_eject": true,
        "error_stop": false,
        "face": "front",
        "types": [
            "CODE128"
        ]
    }
}
```


❏ The current setting values that have been configured with this API are returned.

❏ Each item is the same as the body of the request.

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Validation error	400	validation_error
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Set Scan Setting (Card)

This API is used to configure the settings for scanning cards with photos.
These settings are retained until Disconnect is performed.
If a parameter is not specified, the default value is used.

Request

Command URI

/api/scan/setting/card

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Item name	Type	Required	Default value	Description
face	String		both	Select from the following. • both • front • back
images	Array (Object)			This array contains captured image information objects.
format	String		jpeg	Select from the following. • jpeg: JPEG format normal compression data • tiff: TIFF format CCITT (Group 4) compressed data • bitmap: Bitmap format uncompressed data • tiff256: TIFF format uncompressed data • jpeg_high: JPEG format high compression (size priority) data • jpeg_low: JPEG format low compression (quality priority) data • jtiff: TIFF format JPEG compressed data
type	String		color	Select from the following. • grayscale • color

Item name	Type	Required	Default value	Description
resolution	String		200dpi	Select from the following. • 200dpi • 300dpi • 600dpi
brightness	Numeric		0	-100≤brightness≤100 If 0, brightness is not corrected.
contrast	Numeric		0	-100≤contrast≤100 If 0, contrast is not corrected.
gamma	Numeric		1.0	Select from the following. • 1.0 • 1.8 • 2.2 If 1.0, gamma is not corrected.
light_source	String		RGB	The light source to be used for scan execution is selected. • RGB: Visible light • IR: Infrared light
buzzer	Array (Object)		[]	This array contains buzzer condition objects.
type	String	✓		Select the conditions under which the buzzer sounds. • success: The buzzer sounds if the reading is successful. • error: The buzzer sounds if a read error occurs.
frequency	String	✓		Select from the following. • 440hz • 880hz • 4000hz
count	Numeric	✓		Number of times the buzzer sounds. 1 ≤ count ≤ 3
endorse	Object			This is an endorsement object.
endorse.type	String		none	Select from the following. • none • electronic
endorse.data	Array (Object)		[]	This array contains endorsement printing information objects. It is an electronic endorsement, with text and images attached to scanned data.
font_type	String		sans-serif	Select from the following. • serif • sans-serif • monospace
bold	Boolean		false	Set whether to use bold text.
font_size	Numeric		10	Font size. 4≤font_size≤72 If type is image, ignore this setting.
top_left_x	Numeric	✓		0≤top_left_x≤235
top_left_y	Numeric	✓		0≤top_left_y≤109
width	Numeric	✓		1≤width≤236
height	Numeric	✓		1≤height≤110

Item name	Type	Required	Default value	Description
direction	String		left_to_right	Select from the following. • left_to_right • bottom_to_up • right_to_left • up_to_bottom
type	String	✓		Select from the following. • text • image
contents	String	✓		If type is text, specify a print string. If type is image, specify Base64 encoded image data. Min. 1 character If type is text: Max. 1024 characters If type is image: Max. 10485760 characters The following image formats are supported. • BMP format (Uncompressed image data only) • JPEG format (Baseline DCT, Progressive) • TIFF format (CCITT Group 3/Group 4 compressed data, uncompressed data only)
barcode	Object			This is a barcode reading information object.
enabled	Boolean		false	Set whether to perform barcode reading.
face	String		front	Select from the following. • both • front • back
types	Array (String)		["CODE128"]	Select from the following. (You can select more than one.) • CODE128 • CODABAR • CODE39 • ITF • EAN/JAN • UPC-A • UPC-E

HTTP body(Example)

```
{
   "face": "both",
   "images": [
      {
         "format": "jpeg",
         "type": "grayscale"
      }
   ],
   "resolution": "300dpi"
}
```

- In this example, the following images can be obtained.

No.	Scan face	File format	Image type
1	Front side	JPEG	Grayscale
2	Back side	JPEG	Grayscale

Default value

```
{
   "face": "both",
   "images": [
      {
         "format": "jpeg",
         "type": "color"
      }
   ],
   "resolution": "200dpi",
   "brightness": 0,
   "contrast": 0,
   "gamma": 1.0,
   "light_source": "RGB",
   "buzzer": [],
   "endorse": {
      "type": "none",
      "data": []
   },
   "barcode": {
      "enabled": false,
      "face": "front",
      "types": [
         "CODE128"
      ]
   }
}
```

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body


```
{
   "face": "both",
   "images": [
      {
         "format": "jpeg",
         "type": "grayscale"
      }
   ],
   "resolution": "300dpi",
   "brightness": 0,
   "contrast": 0,
   "gamma": 1.0,
   "light_source": "RGB",
   "buzzer": [],
   "endorse": {
      "type": "none"
   },
   "barcode": {
      "enabled": false,
      "face": "front",
      "types": [
         "CODE128"
      ]
   }
}
```


❏ The current setting values that have been configured with this API are returned.

❏ Each item is the same as the body of the request.

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Validation error	400	validation_error
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Start Scan (Check)

This API is used to start the check scanning process.
The feeder waits for paper feed.

Request

Command URI

/api/scan/start/check

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Item name	Type	Required	Default value	Description
limit	Numeric or null	-	null	Number of scans 0 ≤ limit ≤ 100 If null, it depends on the ASF (Auto Sheet Feeder).
timeout	Numeric or null	-	null	Document insertion wait timeout (seconds) 1 ≤ timeout ≤ 300 If null, the wait time will be infinite.
transaction_number	Numeric or null	-	null	Transaction number 0 ≤ transaction_number ≤ 9999999999999999 If null is specified, the value is 1 the first time after Connect, and the last transaction number +1 each time thereafter. The transaction assigns a number from 0 to 9999999999999999 to the document.
step	Numeric	-	1	Increments the transaction number 1 ≤ step ≤ 10

HTTP body(Example)

```
{
   "limit": 1,
   "timeout": null,
   "transaction_number": null,
   "step": 1
}
```

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Validation error	400	validation_error
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Start Scan (Card)

This API is used to start the photo card scanning process.
The card feeder waits for a card to be inserted.

Request

Command URI

/api/scan/start/card

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Item name	Type	Required	Default value	Description
timeout	Numeric or null	-	null	Document insertion wait timeout (seconds) 1 ≤ timeout ≤ 300 If null, the wait time will be infinite.
transaction_number	Numeric or null	-	null	Transaction number 0 ≤ transaction_number ≤ 9999999999999999 If null is specified, the value is 1 the first time after Connect, and the last transaction number +1 each time thereafter. The transaction assigns a number from 0 to 9999999999999999 to the document.

HTTP body(Example)

```
{
   "timeout": null,
   "transaction_number": null
}
```

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Validation error	400	validation_error
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Cancel Scan

This API is used to stop the running scan process.
Although canceling a scan will stop it immediately, a scan that is still in progress might continue.

Request

Command URI

/api/scan/cancel

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scan operation not in progress	400	not_scanning
Authentication error	401	access_token_verification_failed

Get Document List

This API is used to obtain the current scanner status and the results of the most recent scanning process.
If you want to check the device status or ink status, run the DeviceStatus API instead of this API.
This API is used to retrieve a list of documents that have been scanned.
Images that have been obtained with the GetImage API are excluded from the GetDocumentList response.
If all of the images in a document are obtained with the GetImage API, that document will be excluded from the GetDocumentList response.

Request

Command URI

/api/docs

HTTP Method

GET

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{
   "latest_result":"success",
   "status":"connected",
```

```
"documents": [
  {
    "transaction_number": 1,
    "front": [
      "/api/docs/1/00001_check_front.jpg"
    ],
    "back": [
      "/api/docs/1/00001_check_back.jpg"
    ],
    "micr": {
      "text": "o005575o t123456780t1234567890o",
      "account_number": "1234567890",
      "amount": "``",
      "bank_number": "5678",
      "serial_number": "005575",
      "epc": "",
      "transit_number": "123456780",
      "check_type": 0,
      "country_code": 0,
      "on_us_field": "1234567890o",
      "auxiliaty_on_us_field": "005575"
    },
    "iqa": {
      "under_size": {
        "result": "pass",
        "width": 60,
        "height": 27
      },
      "over_size": {
        "result": "pass",
        "width": 60,
        "height": 27
      },
      "min_compressed_image_size": {
        "result": "pass",
        "size": 117335
      },
      "max_compressed_image_size": {
        "result": "pass",
        "size": 117335
      },
      "front_rear_image_mismatch": {
        "result": "pass",
        "abs_width_diff": 0,
        "abs_height_diff": 0
      },
      "image_too_light": {
        "result": "pass",
        "black_pixels": 0,
        "brightness": 874,
        "contrast": 474
      },
      "image_too_dark": {
        "result": "pass",
        "black_pixels": 0,
        "brightness": 874
      },
      "horizontal_streaks_present": {
        "result": "pass",
        "streak_count": 1,
        "streak_height": 1
      },
```

```
            "excessive_spot_noise": {
              "result": "pass",
              "count": 0
            },
            "image_out_of_focus": {
              "result": "pass",
              "image_focus_score": 100
            },
            "folded_torn_doc_corners": {
              "result": "pass",
              "topleft_width": 0,
              "topleft_height": 0,
              "topright_width": 0,
              "topright_height": 0,
              "bottom_left_width": 0,
              "bottom_left_height": 0,
              "bottom_right_width": 0,
              "bottom_right_height": 0
            },
            "folded_torn_doc_edges": {
              "result": "pass",
              "top_width": 0,
              "top_height": 0,
              "left_width": 0,
              "left_height": 0,
              "right_width": 0,
              "right_height": 0,
              "bottom_width": 0,
              "bottom_height": 0
            },
            "doc_framing_error": {
              "result": "pass",
              "top": 0,
              "left": 0,
              "right": 0,
              "bottom": 0
            },
            "excessive_doc_skew": {
              "result": "pass",
              "angle": -3,
              "range": 0
            },
            "carbon_strip_detection": {
              "result": "pass",
              "strip_height": 0
            },
            "piggy_back": {
              "result": "pass"
            }
          },
          "ocr_text": "ABCDEFGHIJKLMNOPQRSTUVWXYZO923456789",
          "barcode": [
            "ABC DEF GHI"
          ]
        }
    ]
}
```

Item name	Type	Description
latest_result	String	The result of the last scan execution. When ScanStart API is executed, it is cleared to an empty string. success: Successful timeout: Timeout no_docs: Timeout for check insertion canceled: Canceled micr_error: MICR data error image_error: Image data error not_exec: Not executed double_feed: Double-feed detected jammed: Paper jam cover_open: Cover open micr_baddata: MICR character recognition error micr_noise: MICR noise error print_data_length_exceed: Printed image size exceeded the printable area print_image_format_error: Print image format error iqa_not_pass: IQA failure no_barcode: No barcode detected less_checks: Insufficient number of sheets disk_full: Insufficient memory error ink_error: Ink error misinsertion: Paper insertion error device_access_error: Device access error paper_exist: Paper exists in the pathway other_error: Unexpected error
status	Object	Current scanner status. One of the following • connected • scanning • printing
documents	Array (Object)	This is an array of document objects.
transaction_number	Numeric	Transaction number
front	Array (String)	This array contains the URI string for the front image.
back	Array (String)	This array contains the URI string for the back image.
micr	Object	This is an MICR data object.
text	Numeric	Scanned MICR data
account_number	String	AccountNumber property
amount	String	Amount property
bank_number	String	BankNumber property
serial_number	String	SerialNumber property
epc	String	EPC property
transit_number	String	TransitNumber property

Item name	Type	Description
check_type	Numeric	CheckType property
country_code	Numeric	CountryCode property
on_us_field	String	MICR On-US field
auxiliaty_on_us_-field	String	Auxiliaty On-US field
iqa	Object	This is an IQA result object.
under_size	Object	This is an under_size result object.
result	String	• pass • not_pass • not_tested
width	Numeric	Image width (unit: 0.1 inch)
height	Numeric	Image height (unit: 0.1 inch)
over_size	Object	This is an over_size result object.
result	String	• pass • not_pass • not_tested
width	Numeric	Image width (unit: 0.1 inch)
height	Numeric	Image height (unit: 0.1 inch)
min_compressed_im-age_size	Object	This is a min_compressed_image_size result object.
result	String	• pass • not_pass • not_tested
size	Numeric	Compressed image size (units: bytes)
max_compressed_im-age_size	Numeric	This is a max_compressed_image_size result object.
result	String	• pass • not_pass • not_tested
size	Numeric	Compressed image size (units: bytes)
front_rear_image_mis-match	Object	This is a front_rear_image_mismatch result object.
result	String	• pass • not_pass • not_tested
abs_width_diff	Numeric	Difference in image width between front and back (units: 0.1 inch)
abs_height_diff	Numeric	Difference in image height between front and back (units: 0.1 inch)
image_too_light	Object	This is an image_too_light result object.
result	String	• pass • not_pass • not_tested
black_pixels	Numeric	Percentage of black pixels in the image (units: 0.1%)
brightness	Numeric	Image brightness (units: 0.1%)
contrast	Numeric	Image contrast (units: 0.1%)

Item name	Type	Description
image_too_dark	Object	This is an image_too_dark result object.
result	String	• pass • not_pass • not_tested
black_pixels	Numeric	Percentage of black pixels in the image (units: 0.1%)
brightness	Numeric	Image brightness (units: 0.1%)
horizontal_streaks_present	Object	This is a streaks result object.
result	String	• pass • not_pass • not_tested
streak_count	Numeric	Number of black stripes in the binary image
streak_height	Numeric	Width of thickest black stripe
excessive_spot_noise	Object	This is a noise result object.
result	String	• pass • not_pass • not_tested
count	Numeric	Average number of spot noise per square inch
image_out_of_focus	Object	This is a focus result object.
result	String	• pass • not_pass • not_tested
image_focus_score	Numeric	The score is estimated with the following formula: (maximum video gradient)/(gray level dynamic range)*(pixel pitch).
folded_torn_doc_corners	Object	This is a corners result object.
result	String	• pass • not_pass • not_tested
topleft_width	Numeric	Width of the tear/bend in the upper left corner of the image (units: 0.1 inch)
topleft_height	Numeric	Height of the tear/bend in the upper left corner of the image (units: 0.1 inch)
topright_width	Numeric	Width of the tear/bend in the upper right corner of the image (units: 0.1 inch)
topright_height	Numeric	Height of the tear/bend in the upper right corner of the image (units: 0.1 inch)
bottom_left_width	Numeric	Width of the tear/bend in the lower left corner of the image (units: 0.1 inch)
bottom_left_height	Numeric	Height of the tear/bend in the lower left corner of image (units: 0.1 inch)
bottom_right_width	Numeric	Width of the tear/bend in the lower right corner of the image (units: 0.1 inch)
bottom_right_height	Numeric	Height of the tear/bend in the lower right corner of the image (units: 0.1 inch)

Item name	Type	Description
folded_torn_doc_edges	Object	This is an edges result object.
result	String	• pass • not_pass • not_tested
top_width	Numeric	Width of the tear/bend in the upper left corner of the image (units: 0.1 inch)
top_height	Numeric	Height of the tear/bend in the upper left corner of the image (units: 0.1 inch)
left_width	Numeric	Width of the tear/bend in the upper right corner of the image (units: 0.1 inch)
left_height	Numeric	Height of the tear/bend in the upper right corner of the image (units: 0.1 inch)
right_width	Numeric	Width of the tear/bend in the lower left corner of the image (units: 0.1 inch)
right_height	Numeric	Height of the tear/bend in the lower left corner of image (units: 0.1 inch)
bottom_width	Numeric	Width of the tear/bend in the lower right corner of the image (units: 0.1 inch)
bottom_height	Numeric	Height of the tear/bend in the lower right corner of the image (units: 0.1 inch)
doc_framing_error	Object	This is a framing result object.
result	String	• pass • not_pass • not_tested
top	Numeric	Width of the top margin of the image (units: 0.1 inch)
left	Numeric	Width of the left margin of the image (units: 0.1 inch)
right	Numeric	Width of the right margin of the image (units: 0.1 inch)
bottom	Numeric	Width of the bottom margin of the image (units: 0.1 inch)
excessive_doc_skew	Object	This is a skew result object.
result	String	• pass • not_pass • not_tested
angle	Numeric	Inclination angle (units: 0.1 degree)
range	Numeric	Fixed to 0: Value indicating that the angle is in the range of -90 to +90
carbon_strip_detection	Object	This is a carbon result object.
result	String	• pass • not_pass • not_tested
strip_height	Numeric	Length of carbon strip (units: 0.1 inch)
piggy_back	Object	This is a piggy_back result object.
result	String	• pass • not_pass • not_tested
ocr_text	String	OCR AB reading result string
barcode	Array (String)	This array contains the barcode reading result string.

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Authentication error	401	access_token_verification_failed

Delete All Documents

This API is used to delete all scan result documents.

Request

Command URI

/api/docs

HTTP Method

DELETE

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Authentication error	401	access_token_verification_failed

Get Image

This API is used to obtain scan result image data.

Images that have been obtained with this API are excluded from the GetDocumentList response.

If all of the images in a document are obtained with this API, that document will be excluded from the GetDocumentList response.

Request

Command URI

/api/docs/{transaction_number}/{image_file_name}

Item name	Required	Value
transaction_number	✓	Transaction number of the image to be obtained
image_file_name	✓	File name of the image to be obtained

HTTP Method

GET

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	{Different for each image format}
Content-Disposition	attachment; filename="{Filename}"

Content-Type by image format

Image type	Content-Type
JPEG	image/jpeg
TIFF	image/tiff
Bitmap	image/bmp

Status code

200

HTTP body

Image data

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Authentication error	401	access_token_verification_failed
Could not find the specified image	404	not_found

Print Cut Sheet

This API is used to start cut-sheet printing.

Request

Command URI

/api/print/cut_sheet

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Item name	Type	Required	Default value	Description
limit	Numeric		1	Maximum number of sheets printed 1 ≤ limit ≤ 100
timeout	Numeric or null		null	Timeout waiting for sheet insertion (seconds). 1≤timeout≤300 If null, the wait time will be infinite.
transaction_number	Numeric or null		null	Transaction number 0 ≤ transaction_number ≤ 9999999999999999 If null is specified, the value is 1 the first time after Connect, and the last transaction number +1 each time thereafter. The transaction assigns a number from 0 to 9999999999999999 to the document.
step	Numeric		1	Increments the transaction number 1 ≤ step ≤ 10

Item name	Type	Required	Default value	Description
data	Array (Object)	✓		This array contains cut-sheet printing information objects.
font_type	String		sans-serif	Select from the following. • font_a: The FontA with 1 unit horizontal and 1 vertical • font_a_w2: The FontA with 2 units horizontal and 1 vertical • font_a_h2: The FontA with 1 unit horizontal and 2 vertical • font_a_w2_h2: The FontA with 2 units horizontal and 2 vertical • font_b: The FontB with 1 unit horizontal and 1 vertical • font_b_w2: The FontB with 2 units horizontal and 1 vertical • font_b_h2: The FontB with 1 unit horizontal and 2 vertical • font_b_w2_h2: The FontB with 2 units horizontal and 2 vertical • serif • sans-serif • monospace
bold	Boolean		false	Set whether to use bold text.
font_size	Numeric		10	Font size. 4 ≤ font_size ≤ 72 If type is image, ignore this setting. Ignore this setting if font_type is font_a or font_b.
top_left_x	Numeric	✓		X-coordinates for print origin position (units: mm) 0 ≤ top_left_x ≤ 214
top_left_y	Numeric	✓		Y-coordinates for print origin position (units: mm) 0 ≤ top_left_y ≤ 49
width	Numeric	✓		Print width (units: mm) 1 ≤ width ≤ 215
height	Numeric	✓		Print height (units: mm) 1 ≤ height ≤ 50
direction	String	✓	left_to_right	Select from the following. • left_to_right • bottom_to_up • right_to_left • up_to_bottom Ignore if data.type set
type	String	✓		Select from the following. • text • image

Item name		Type	Required	Default value	Description
	contents	String	✓	0	If type is text, specify a print string. If type is image, specify Base64 encoded image data. Min. 1 character If type is text: Max. 1024 characters If type is image: Max. 10485760 characters The following image formats are supported. • BMP format (Uncompressed image data only) • JPEG format (Baseline DCT, Progressive) • TIFF format (CCITT Group 3/Group 4 compressed data, uncompressed data only)

HTTP body(Example)

Print "Hello world!" on a single cut sheet.

```
{
   "limit": 1,
   "timeout": null,
   "transaction_number": null,
   "step": 1,
   "data": [
      {
         "font_type": "sans-serif",
         "bold": true,
         "font_size": 10,
         "top_left_x": 0,
         "top_left_y": 0,
         "width": 50,
         "height": 10,
         "direction": "left_to_right",
         "type": "text",
         "contents": "Hello world!"
      }
   ]
}
```

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Validation error	400	validation_error
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Get Print Status

This API is used to obtain the current scanner status and the results of the most recent print process.
If you want to check the device status or ink status, run the DeviceStatus API instead of this API.

Request

Command URI

/api/print/status

HTTP Method

GET

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{
   "latest_result":"success",
   "status":"connected"
}
```

Item name	Type	Description
latest_result	String	Result of the last print job that was executed. If the print start API is executed, this string is cleared to a null character. Null character or error code string success: Successful timeout: Timeout no_docs: Timeout for check insertion canceled: Canceled not_exec: Not executed double_feed: Double-feed detected jammed: Paper jam cover_open: Cover open print_data_length_exceed: Printed image size exceeded the printable area print_image_format_error: Print image format error ink_error: Ink error other_error: Unexpected error
status	Object	Current scanner status. One of the following: • connected • scanning • printing

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Authentication error	401	access_token_verification_failed

Cancel Print

This API is used to stop cut-sheet printing.

Request

Command URI

/api/print/cancel

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Printing not in progress	400	not_printing
Authentication error	401	access_token_verification_failed

Get Device Status

This API is used to obtain the device status.

Request

Command URI

/api/device/status

HTTP Method

GET

HTTP header

None

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{
   "device_status": ["ok"],
   "ink_status": ["ok"]
}
```

Item name	Type	Description	
device_status	Array (String)	This array contains device status strings.	
		ok	No error
		no_response	No response from the device
		off_line	Offline status
		cover_open	Cover open
		wait_eject	Waiting for ejection
		mechanical_err	Recoverable error
		autocutter_err*	Auto-cutter error
		unrecover_err	Unrecoverable error
		autorecover_err	Automatic recovery error
		receipt_near_end*	Roll paper near end
		receipt_end*	Roll paper end
		wait_insert	Waiting for insertion
		paper_detected	Paper detected
		card_detected	Card detected
ink_status	Array (String)	This array contains ink status strings.	
		ok	No error
		near_end	Ink level low
		replace_cartridge	Replace ink cartridge
		no_cartridge	Ink cartridge not present
		cleaning	Head cleaning in progress
		busy	Scanner operation in progress

* TM-S2000II-NW not supported

Error cause / Status code / Error code string

None

Reset Device

This API is used to reset the main unit of the scanner.

Request

Command URI

/api/device/reset

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Get Log

This API is used to obtain the log of the main unit of the scanner.

Request

Command URI

/api/device/log

HTTP Method

GET

HTTP header

None

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/zip

Status code

200

HTTP body

ZIP file containing log data

MICR Cleaning

This API is used to perform MICR cleaning of the scanner.

Request

Command URI

/api/device/cleaning/micr

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Timeout while waiting for cleaning paper insertion	400	timeout
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Head Cleaning

This API is used to perform ink head cleaning on the scanner.

Request

Command URI

/api/device/cleaning/head

HTTP Method

POST

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Get Maintenance Counter

This API is used to obtain maintenance counter information.

Request

Command URI

/api/device/counter

HTTP Method

GET

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body


```
{
   "resettable": {
      "count_of_thermal_head_energization": 0,
      "number_of_fed_by_thermal_head": 0,
      "number_of_fed_for_roll_paper": 0,
      "number_of_ij_head_shots_column_a": 0,
      "number_of_ij_head_shots_column_b": 0,
      "count_of_pump_motor_operations": 0,
      "count_of_autocutter_drive": 0,
      "count_of_magnetic_ink_character_read": 0,
      "count_of_check_paper_scanning": 0,
      "count_of_card_scanning": 0,
      "count_of_check_paper_feeding": 0,
      "duration_of_product_operation": 0,
      "count_of_hopper_open_close": 0,
      "count_of_pocket_switch": 0
   },
   "cumulative": {
      "count_of_thermal_head_energization": 0,
      "number_of_fed_by_thermal_head": 0,
      "number_of_fed_for_roll_paper": 0,
      "number_of_ij_head_shots_column_a": 0,
      "number_of_ij_head_shots_column_b": 0,
      "count_of_pump_motor_operations": 0,
      "count_of_autocutter_drive": 0,
      "count_of_magnetic_ink_character_read": 0,
      "count_of_check_paper_scanning": 0,
      "count_of_card_scanning": 0,
      "count_of_check_paper_feeding": 0,
      "duration_of_product_operation": 0,
      "count_of_hopper_open_close": 0,
      "count_of_pocket_switch": 0
   }
}
```


Item name	Type	Description	
resettable	Object	This is a maintenance_counter object that can be reset.	
		count_of_thermal_head_energization[*1]	Number of thermal head timing pulse [count]
		number_of_fed_by_thermal_head[*1]	Number of lines fed by thermal head [lines]
		number_of_fed_for_roll_paper[*1]	Number of lines fed for roll paper [lines]
		number_of_ij_head_shots_column_a	Number of IJ head shots (Column A) [units: 1000 shots]
		number_of_ij_head_shots_column_b	Number of IJ head shots (Column B) [units: 1000 shots]
		count_of_pump_motor_operations	Count of pump operations [count]
		count_of_autocutter_drive[*1]	Number of autocutter operations [times]
		count_of_magnetic_ink_character_read	Count of magnetic ink character read [count]
		count_of_check_paper_scanning	Count of cut sheet scanning [count]
		count_of_card_scanning	Count of card scanning [count]
		count_of_check_paper_feeding	Count of ASF feeding [count]
		duration_of_product_operation	Duration of product operation [hours]
		count_of_hopper_open_close	Count of hopper open/close [count]
		count_of_pocket_switch	Count of pocket switch [count]

Item name	Type	Description	
cumulative	Object	This is a cumulative maintenance_counter object that cannot be reset.	
		count_of_thermal_head_energization[*1]	Number of thermal head timing pulse [count]
		number_of_fed_by_thermal_head[*1]	Number of lines fed by thermal head [lines]
		number_of_fed_for_roll_paper[*1]	Number of lines fed for roll paper [lines]
		number_of_ij_head_shots_column_a	Number of IJ head shots (Column A) [units: 1000 shots]
		number_of_ij_head_shots_column_b	Number of IJ head shots (Column B) [units: 1000 shots]
		count_of_pump_motor_operations	Count of pump operations [count]
		count_of_autocutter_drive[*1]	Number of autocutter operations [times]
		count_of_magnetic_ink_character_read	Count of magnetic ink character read [count]
		count_of_check_paper_scanning	Count of cut sheet scanning [count]
		count_of_card_scanning	Count of card scanning [count]
		count_of_check_paper_feeding	Count of ASF feeding [count]
		duration_of_product_operation	Duration of product operation [hours]
		count_of_hopper_open_close	Count of hopper open/close [count]
		count_of_pocket_switch	Count of pocket switch [count]

*1 This is for TM-S9000II-NW only.

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Reset Maintenance Counter

This API is used to reset maintenance counter information.

Use a path parameter to specify the item to be reset.

Request

Command URI

/api/device/counter/{counter_name}

Item name	Required	Value	
counter_name	✓	Counter name to be reset	
		count_of_thermal_head_energization*1	Number of thermal head timing pulse [count]
		number_of_fed_by_thermal_head*1	Number of lines fed by thermal head [lines]
		number_of_fed_for_roll_paper*1	Number of lines fed for roll paper [lines]
		number_of_ij_head_shots_column_a	Number of IJ head shots (Column A) [units: 1000 shots]
		number_of_ij_head_shots_column_b	Number of IJ head shots (Column B) [units: 1000 shots]
		count_of_pump_motor_operations	Count of pump operations [count]
		count_of_autocutter_drive*1	Number of autocutter operations [times]
		count_of_magnetic_ink_character_read	Count of magnetic ink character read [count]
		count_of_check_paper_scanning	Count of cut sheet scanning [count]
		count_of_card_scanning	Count of card scanning [count]
		count_of_check_paper_feeding	Count of ASF feeding [count]
		duration_of_product_operation	Duration of product operation [hours]
		count_of_hopper_open_close	Count of hopper open/close [count]
		count_of_pocket_switch	Count of pocket switch [count]

*1 This is for TM-S9000II-NW only.

HTTP Method

DELETE

HTTP header

Header name	Required	Value
Authorization	✓	Specify the token string obtained with the Connect API.

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Get Default Scan Setting (Check)

This API is used to obtain the default scan setting values (check).

Request

Command URI

/api/scan/setting/check

HTTP Method

GET

HTTP header

None

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

Refer to the POST body of the Set Scan Setting API (Check). (page 29)

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Get Default Scan Setting (Card)

This API is used to retrieve the stored default scan setting values (card).

Request

Command URI

/api/scan/setting/card

HTTP Method

GET

HTTP header

None

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

Refer to the POST body of the Set Scan Setting API (Card). (page 44)

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Save Default Scan Setting (Check)

This API is used to register the default scan settings for checks.
The initial value that is registered with this API is retained even after the device is disconnected or turned off.

Request

Command URI

/api/scan/setting/check

HTTP Method

PUT

HTTP header

Header name	Required	Value
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Refer to the POST body of the Set Scan Setting API (Check). (page 29)

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Validation error	400	validation_error
Authentication error	401	access_token_verification_failed

Save Default Scan Setting (Card)

This API is used to register the default scan settings for cards.
The initial value that is registered with this API is retained even after the device is disconnected or turned off.

Request

Command URI

/api/scan/setting/card

HTTP Method

PUT

HTTP header

Header name	Required	Value
Content-Type	✓	application/json; charset=utf-8

Query string

None

HTTP body

Refer to the POST body of the Set Scan Setting API (Card). (page 44)

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Validation error	400	validation_error
Authentication error	401	access_token_verification_failed

Reset Default Scan Setting (Check)

This API is used to reset the registered default scan settings for checks to the factory settings.

Request

Command URI

/api/scan/setting/check

HTTP Method

DELETE

HTTP header

None

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

Reset Default Scan Setting (Card)

This API is used to reset the registered default scan settings for cards to the factory settings.

Request

Command URI

/api/scan/setting/card

HTTP Method

DELETE

HTTP header

None

Query string

None

HTTP body

None

Response - Normal Response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Status code

200

HTTP body

```
{}
```

Response - Error response

HTTP header

Header name	Value
Content-Type	application/json; charset=utf-8

Error cause / Status code / Error code string

Error cause	Status code	Error code string
Scanner operation in progress	400	device_busy
Authentication error	401	access_token_verification_failed

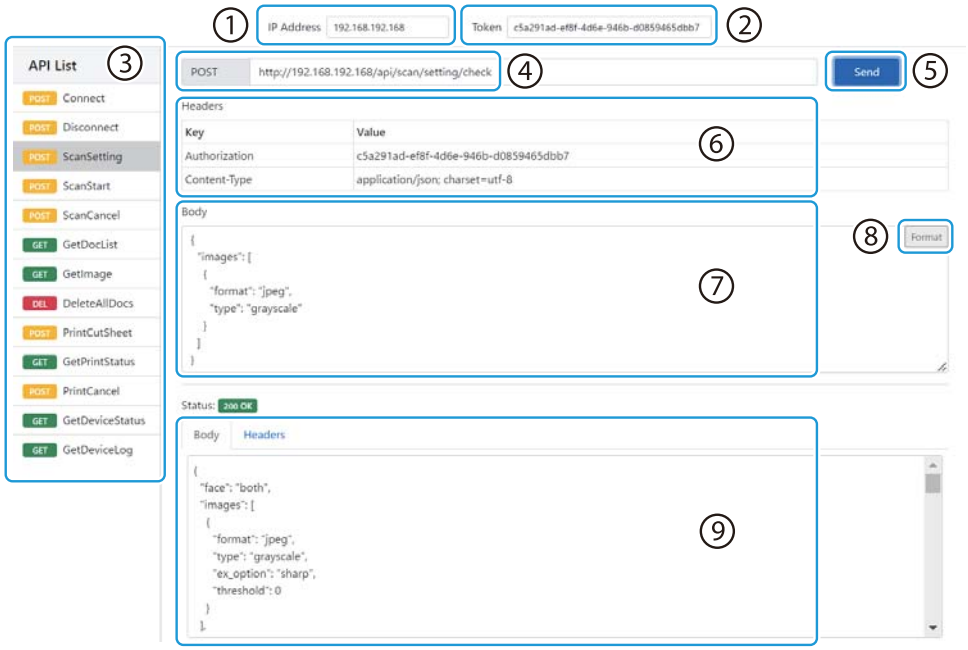
Scan Web API Sample Program

Overview

The sample program uses Web API to scan a check, obtain the image and MICR data, and print them on a cut-sheet.

The data to be sent and received is displayed here, to help customers create their applications.

Screen



No.	Name	Description
1	IP Address	Enter the IP address of the scanner to connect to.
2	Access Token	Displays the access token obtained via the Connect API.
3	API List	Displays a list of APIs that can be executed with the sample program. The content of (4), (6), and (7) will change according to the selected API.
4	URL	Displays the HTTP method for the API and the URL to request.
5	Send button	Executes the API.
6	Request Header	Displays the request header for the API.
7	Request Body	Enter the request body for the API in JSON format. The color of the border around the input field indicates the following: Blue: The entered content is correct. Red: The entered content does not match JSON format.
8	Format	Formats the JSON in the input field so that it is easy to read.
9	Response informations	Displays the response content after the API is executed. This will be displayed once the API has been executed.

Operating Environment

Web Browser

- Mozilla Firefox 91 or later
- Google Chrome 89 or later
- Microsoft Edge 91 or later
- Safari on iOS 13 or later

* Windows Internet Explorer is not supported.

Function

The sample program uses the following APIs.

- Functions provided by the sample program

API Name	Methond	URI
Connect	POST	/api/connect
Disconnect	POST	/api/disconnect
ScanSetting(Check)	POST	/api/scan/setting/check
ScanStart(Check)	POST	/api/scan/start/check
ScanCancel	POST	/api/scan/cancel
GetDocumentList	GET	/api/docs
DeleteAllDocuments	DELETE	/api/docs
GetImage	GET	/api/docs/{transaction_number}/{image_file_name}
PrintCutSheet	POST	/api/print/cut_sheet
GetPrintStatus	GET	/api/print/status
PrintCancel	POST	/api/print/cancel
GetDeviceStatus	GET	/api/device/status
GetDeviceLog	GET	/api/device/log

- Functions not provided by the sample program

API Name	Methond	URI
KeepAlive	POST	/api/keepalive
ScanSetting(Card)	POST	/api/scan/setting/card
ScanStart(Card)	POST	/api/scan/start/card
DeviceReset	POST	/api/device/reset
MICRCleaning	POST	/api/device/cleaning/micr
HeadCleaning	POST	/api/device/cleaning/head
GetCounter	GET	/api/device/counter
ResetCounter	DELETE	/api/device/counter/{counter_name}
GetDefaultScanSetting(Check)	GET	/api/scan/setting/check
GetDefaultScanSetting(Card)	GET	/api/scan/setting/card
SaveDefaultScanSetting(Check)	PUT	/api/scan/setting/check
SaveDefaultScanSetting(Card)	PUT	/api/scan/setting/card
ResetDefaultScanSetting(Check)	DELETE	/api/scan/setting/check
ResetDefaultScanSetting(Card)	DELETE	/api/scan/setting/card

Using the Sample Program

The following operations are described in this document.

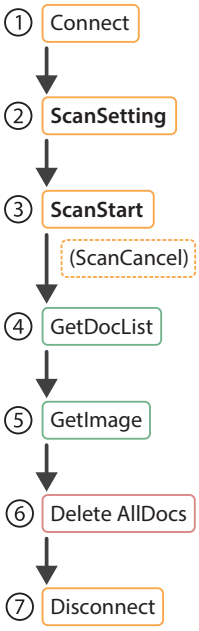
- Connecting, scanning a check, printing a cut-sheet, and disconnecting

Work Flow

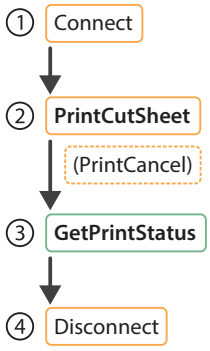
The work flow for the sample program is shown below.

- When you want to obtain a check image and MICR data → Scan a check
- When you want to print on cut-sheets (single-sheet receipts) → Print a cut-sheet
- When you want to obtain the printer status → Obtain a device (printer: StatusAPI)
- When analyzing errors → Obtain printer log

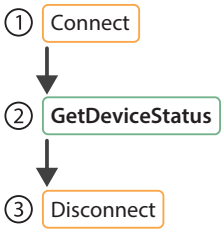
● Scan a check



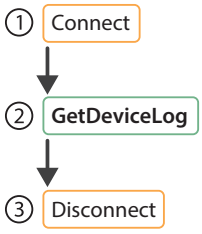
● Print a cut-sheet



● Obtain a device (printer: StatusAPI)



● Obtain printer log

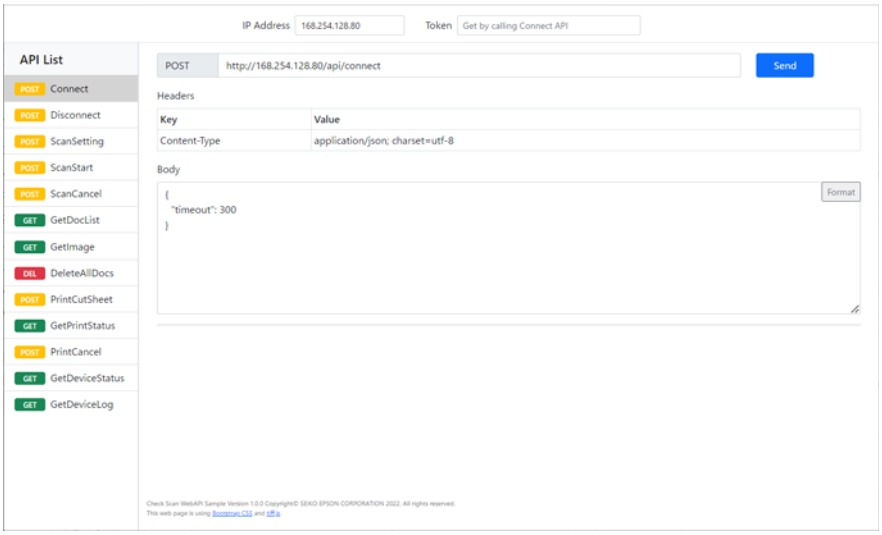


Starting the Sample Program

Use the following steps to start the sample program.

1 **Extract the sample program (scan-webapi-sample_v1.0.0.zip) in a folder of your choice.**

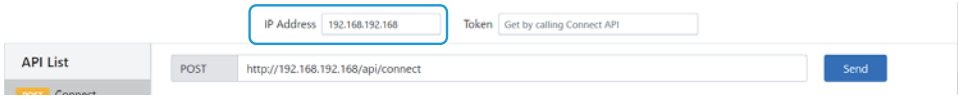
2 **Open index.html in your browser. The following screen is displayed.**



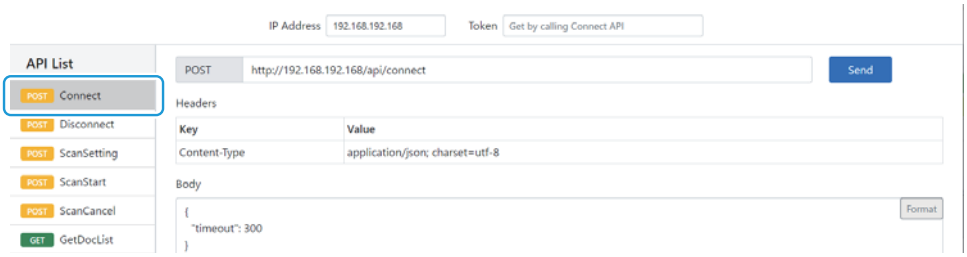
Connecting

Start the use of the scanner. The Connect API is used to gain exclusive access to the scanner. No other programs or devices will be able to use the scanner while this device is connected. When the timeout period specified in the parameters is exceeded, the connection is automatically disconnected and any documents that have been scanned are deleted.

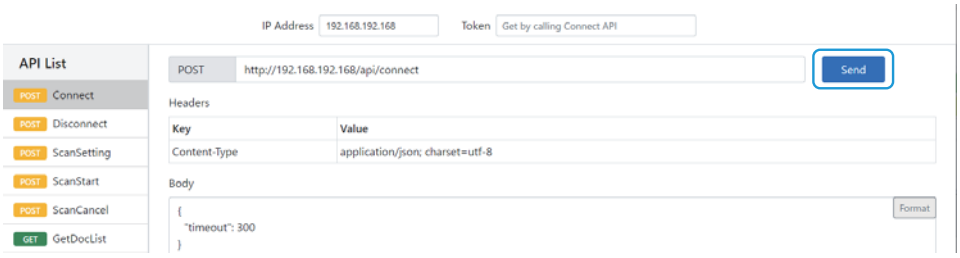
1 Enter the IP address of the scanner.



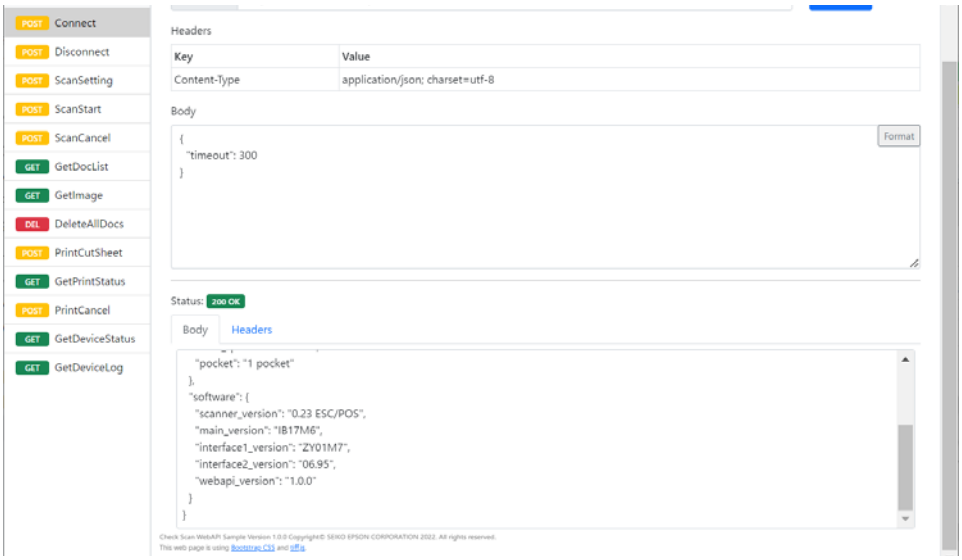
2 Select [Connect] in the [API List].



3 Click the [Send] button to execute the Connect API.



The result of executing the API is displayed.

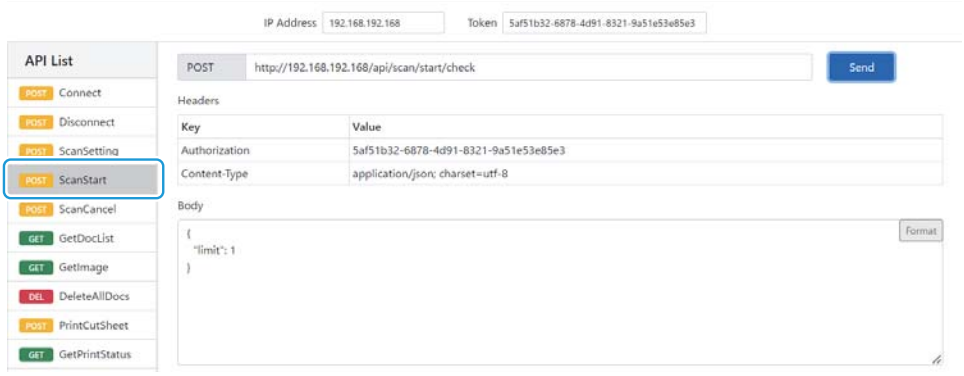


Scanning a Check

This section describes how to scan a check to obtain image data. To configure the scan settings, first execute ScanSetting and then ScanStart.

1 **Make sure the scanner is connected.**

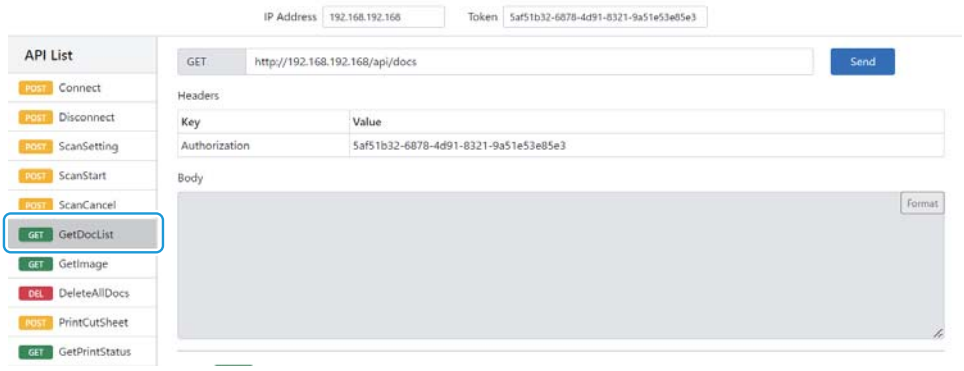
2 **Select [ScanStart] in the [API List].**



3 **Click the [Send] button to execute the ScanStart API.**

4 **The scanner waits for a check to be inserted. When a check is inserted, scanning is performed.**

5 **Select [GetDocList] in the [API List].**



6 **Click the [Send] button to execute the GetDocumentList API.**

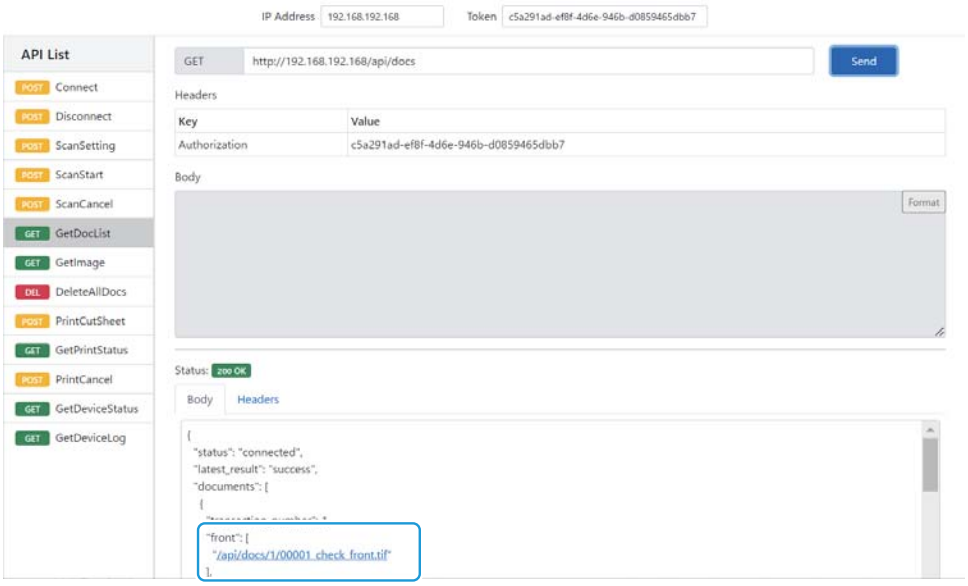
7 The scan results are displayed.

In the "documents" section, a "transaction_number" is assigned to each check. The "front" and "back" sections contain the URLs of the images, and the "micr" section contains the MICR data.


```
{
   "status": "connected",
   "latest_result": "canceled",
   "documents": [
      {
         "transaction_number": 22,
         "front": [
            "/api/docs/22/00022_check_front.jpg"
         ],
         "back": [
            "/api/docs/22/00022_check_back.jpg"
         ],
         "micr": {
            "text": "t123456780t1234567890o  1309",
            "account_number": "1234567890",
            "amount": "",
            "bank_number": "5678",
            "serial_number": "1309",
            "epc": "",
            "transit_number": "123456780",
            "check_type": 0,
            "country_code": 0,
            "on_us_field": "1234567890o  1309",
            "auxiliaty_on_us_field": ""
         }
      },
      {
         "transaction_number": 23,
```

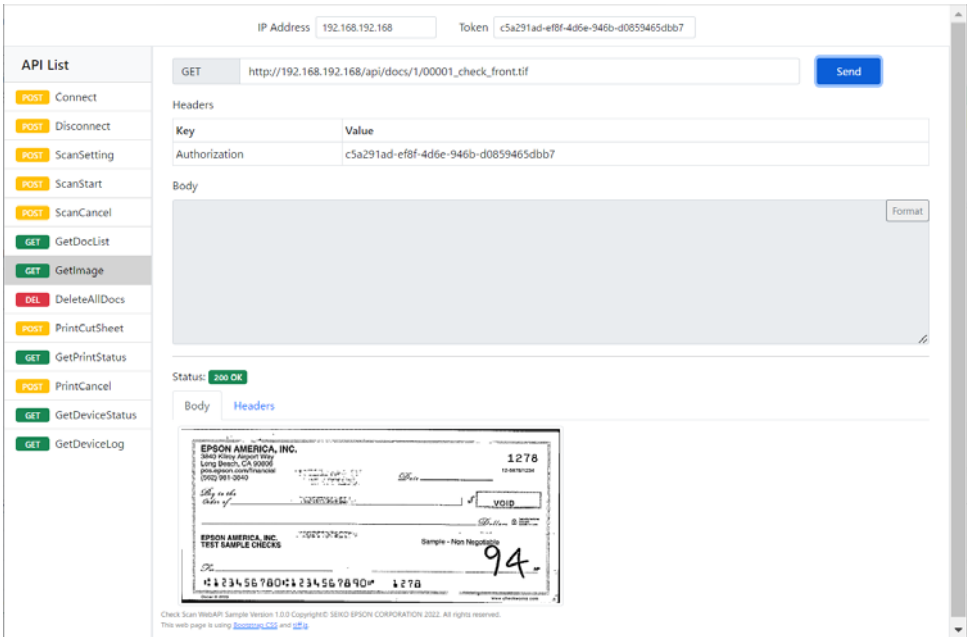

8 Select a URL for an image in the results.

When you select a URL, the GetImage API in the API list will be selected.



9 **Click the [Send] button to execute the GetImage API.**

The specified image data is displayed.



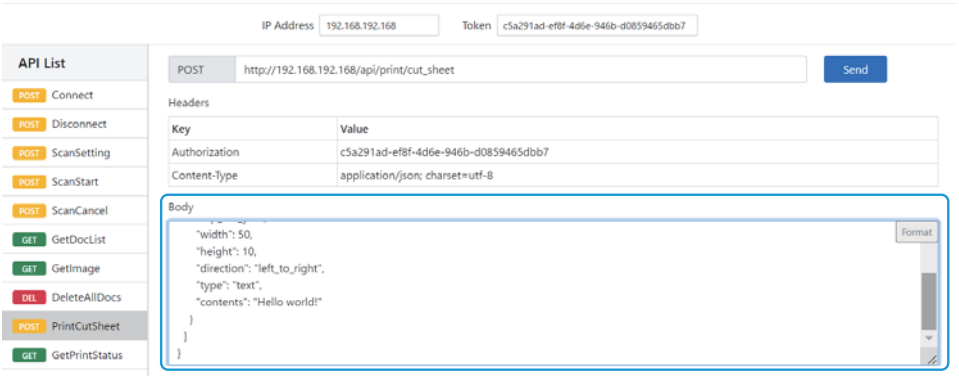
10 **When you have obtained the image, delete the data in the scanner. Select [DeleteAllDocs] in the [API List].**

11 **Click the [Send] button to execute the DeleteAllDocs API.**

Printing a Cut-Sheet

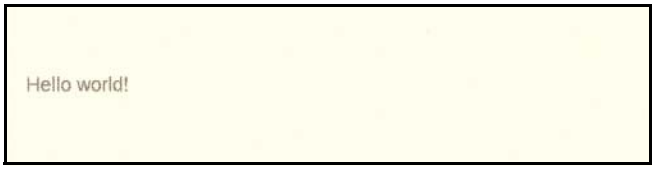
1 **Make sure the scanner is connected.**

2 **Select [PrintCutSheet] in the [API List]. Edit the Body for the content to be printed.**



3 **Click the [Send] button to execute the PrintCutSheet API.**

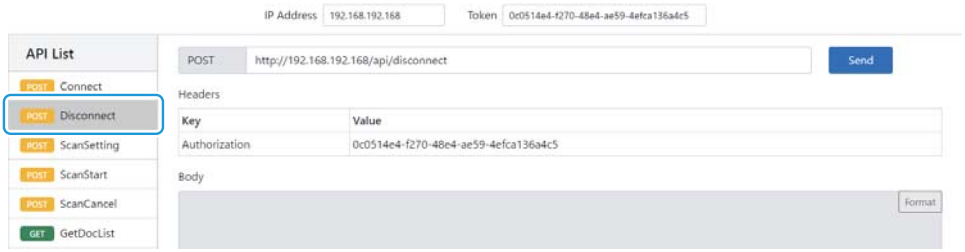
When a cut sheet is loaded, the job is printed.



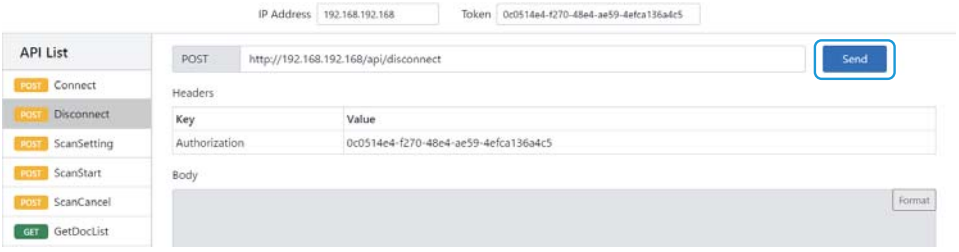
Disconnecting

1 **Make sure the scanner is connected.**

2 **Select [Disconnect] in the [API List]. Edit the Body for the content to be printed.**



3 **Click the [Send] button to execute the Disconnect API.**



Overview of ePOS-Print XML

This chapter describes the features of and the specifications for ePOS-Print.

Overview

ePOS-Print is an Epson's unique printing function that uses XML and Web Service.

This function creates request messages in XML format using the application on devices such as computers, smart phones, and tablets, and performs printing to a TM printer on the network using Web Service.

This document is made up of the following sections:

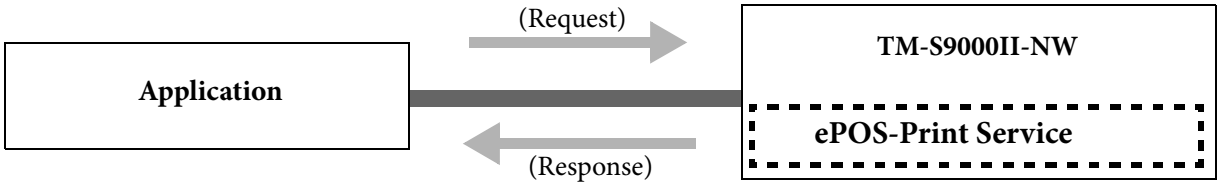
- XML Print Service (p.111)
- XML Reference (p.114)

For details about the provided articles, refer to page 115.

XML Print Service

ePOS-Print XML allows printing to TM-S9000II-NW from applications in an environment or on OS where HTTP communication is possible. ePOS-Print XML embedded into native applications on smart phones and tablet terminals makes it possible to print from such devices.

When a print document is sent from a host to an TM-S9000II-NW using SOAP/HTTP, the TM-S9000II-NW performs printing and returns a response document.



Features

- ❏ Allows printing from SOAP/HTTP-supported devices. Not OS-dependent.
- ❏ Installation of drivers and plug-ins is not required.
- ❏ No computers or servers are required for printing.
- ❏ Automatically checks the status of the TM printer before printing. There is no need for checking the status of the TM printer in advance.
- ❏ Does not respond to a printer's function to automatically send its status (Auto Status Back). Instead, capable of sending an empty print command and checking the status of the TM printer based on the result of command transmission.
- ❏ To change the printer settings, utility dedicated to each printer or other utility programs should be used.
- ❏ Capable of obtaining the printer status even when the printer is offline.

Interface Type

❏ SOAP/HTTP Interface (p.112)

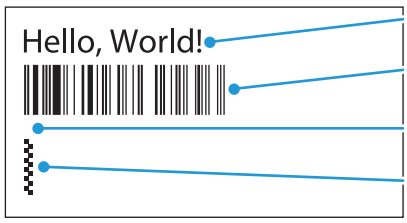
SOAP/HTTP Interface

When a print document is sent from an application to ePOS-Print Service using SOAP/HTTP, ePOS-Print Service performs printing and returns a response document.

Specification

Item	Description
End point address	http://[IP address of the TM-S9000II-NW]/cgi-bin/epos/service.cgi?timeout=[timeout time]
Request message	A SOAP message is sent against a HTTP POST method request. One print document is specified in the SOAP body.
Response message	A SOAP message is returned against a HTTP POST method response. A response document is put in the SOAP body.

Print Image



{Print text}

{Print barcode: EAN13}

{Paper feed: 24 dots}

{Print raster image}

Operating Environment

Application Environment

An environment in which HTTP communication is possible and XML documents can be handled

- ❏ OS (Example: iOS/ Android™/ Windows CE/ Windows/ Linux/ Mac OS, etc.)
- ❏ HTML5-supported browser, Flex
- ❏ Java, .NET Framework

Device

A device capable of HTTP communication by connecting to the network.

(Example: smart phones, tablet terminals, computers, etc.)

Development Environment

An editor program capable of editing XML documents.

(Example: Visual Studio/Eclipse, etc.)

Supported Printer

- ❏ TM-S9000II-NW

Restrictions

- ❏ When multiple tones are set for raster images, intermitting printing may occur because the amount of data to print increases and white stripes may appear in the print result.
- ❏ The scan quality of barcodes/2D-codes printed as multiple-tone raster images cannot be guaranteed. Print them as two-tone images.
- ❏ Messages that contain BOM are not supported.
- ❏ Only UTF-8 character encoding is supported.

XML Reference

ePOS-Print XML is a POS printer command system that defines the major features of the ESC/POS command system using XML.

XML Schemas

The ePOS-Print XML syntax is defined by W3C XML Schemas.

XML schemas help applications strictly validate XML documents using the XML API of the operating environment. In addition, application developers can efficiently create XML documents using the XML document editing support function of the integrated development environment.

File name : epos-print-x.x.x.xsd

Namespace

The ePOS-Print XML namespaces and namespace identifiers used in this document are shown as follows:

Identifier	Namespace
(Default)	http://www.epson-pos.com/schemas/2011/03/epos-print
tns	http://www.epson-pos.com/schemas/2011/03/epos-print
xs	http://www.w3.org/2001/XMLSchema

Command Type

Item	Description
Common	Alignment, Rotation, Line Spacing, Page mode
Character	ANK
Character style	Font, Smoothing, Double-width, Double-height , Horizontal scale, Vertical scale, White/Black Reverse, Underline, Emphasized Color, Print position
Paper feed	By Unit, By Line
Raster image	Mono, Multiple-tone
NV logo	Mono, Multiple-tone
Barcode	UPC-A, UPC-E, EAN13(JAN13), EAN8(JAN13), CODE39, ITF,CODABAR(NW-7), CODE93, CODE128, GS1-128, GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 DataBar Expanded
2D-Code	PDF417, QR Code, MaxiCode, GS1 DataBar
Paper cut	No Feed, Feed Cut, Cut Reservation
Drawer Kick-out	Connector, ON Time
Layout	Paper layout setting
Command	ESC/POS insertion
Response	Print result, Error code, Status
Recovery from an error	Recovery from errors that can be recovered from, Buffer clear
Reset	Printer reset

Provided Article

Package

File name	Description
TM-S9000II-NW_2000II-NW_webapi_rg_en_revx.pdf	This Document.

Manual

- ❏ TM-S9000II-NW/TM-S2000II-NW Web API Reference Guide (This Document)
- ❏ TM-S9000II-NW Technical Reference Guide

Sample Program

ePOS-Print_Sample_XML_Vx.x.xE.zip

Contains the following:

- ❏ epos-print-x.x.x.xsd (ePOS-Print XML schemas)
- ❏ editor/index.html (ePOS-Print Editor)

Download

- ❏ For customers in North America, go to the following web site:
  https://www.epson.com/support/
- ❏ For customers in other countries, go to the following web site:
  https://epson.sn

Programming Guide for ePOS-Print XML

This chapter describes how to write programs in the application development using ePOS-Print.

Executing Print Service

NOTE	In this chapter, descriptions are made mainly based on a HTML5 environment. For an environment other than the above, read such descriptions in ways that suit the relevant environment.

Printing Mode of the TM-S9000II-NW

The TM-S9000II-NW has two printing modes: standard mode and page mode.

Standard mode

In standard mode, characters are printed line by line. The line feed space is adjusted based on the font size and the height of images, barcodes, etc. This mode is suitable for the type of printing such as printing receipts that requires the paper length to change according to the print space.

Page mode

In page mode, you set a print area, lay out data in it, and print the data in a batch operation. Characters, images, and barcodes are laid out in the print positions (coordinates).

Programming Flow

For the ePOS-Print XML, programming is performed based on the following work flow:

1. Print Document Creation (page 117)

- ❏ To create a text print document: (page 118)
- ❏ To create a graphic print document: (page 119)
- ❏ To create a page mode print document: (page 120)

↓

2. Transmission of Print Document (page 121)

↓

3. Reception of Print Result (page 123)

NOTE	To ensure successful print operation, write a program in such a way that data is sent after checking the printer status. For the above procedure, refer to "Checking the Printer Status" on page 127.

Print Document Creation

Create a print document using elements in the epos-print root element. For details on elements, refer to "ePOS-Print XML Reference" on page 128.

Refer to the following program for print document creation.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>TITLE</title>
<script type="text/javascript">
   function createDocument() {
      //Start print document creation.
      var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                    print">';

      //Create a print document
      request += '<text lang="en" smooth="true"/>';
      request += '<text font="font_a"/>';
      request += '<text width="3" height="3">Hello, World!&#10;</text>';
      request += '<cut type="feed"/>';

      //End print document creation.
      request += '</epos-print>';

   }
</script>
</head>
<body>
</body>
</html>
```

To create a text print document:

Create a text print document using the text element in the epos-print root element. Configure the settings for the text to print using the attributes of the text element.

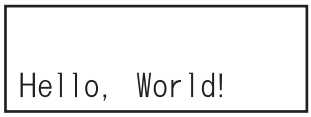
Refer to the following program for print document creation.

For the string "Hello World!", to create a print document based on the following settings:

- ❏ Font: FontA
- ❏ Scale: x 4 (horizontal) and x 4 (vertical)
- ❏ Style: Bold

```
<script type="text/javascript">
   function createDocument() {
      //Start print document creation.
      var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                    print">';

      //Create a print document
      //<Configure the print character settings>
      request += '<text lang="en"/>';
      request += '<text smooth="true"/>';
      request += '<text font="font_a"/>';
      request += '<text width="4" height="4"/>';
      request += '<text em="true"/>';

      //<Specify the character string to print>
      request += '<text>Hello, World!&#10;</text>';

      //<Specify the feed cut>
      request += '<cut type="feed"/>';

      //End print document creation.
      request += '</epos-print>';

   }
</script>
```

Print image



To create a graphic print document:

Create a graphic print document using the image element in the epos-print root element. Set the image size using an attribute of the image element. For graphics, specify raster graphic bit-image data using a base 64 encoded character string.

Refer to the following program for print document creation.

To create a print document with a checkered flag raster image:

```
<script type="text/javascript">
   function createDocument() {
      //Start print document creation.
      var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                     print">';

      //Create a print document
      //<Specify the raster image>
      request += '<image width="8" height="48">8PDw8A8PDw/w8PDwDw8PD/
                  Dw8PAPDw8P8PDw8A8PDw/w8PDwDw8PD/Dw8PAPDw8P</image>';

      //<Specify the feed cut>
      request += '<cut type="feed"/>';

      //End print document creation.
      request += '</epos-print>';

   }
</script>
```

NOTE	• This section describes how to print a raster image. In addition, there is also a method of printing graphics registered in the NV memory of the printer. For details, refer to "<logo>" on page 142. • For details on how to create raster images, refer to "Encoding Graphic Data" on page 172.

Print image

To create a page mode print document:

Create a graphic print document using the image element in the epos-print root element. Set the image size using an attribute of the image element. For graphics, specify raster graphic bit-image data using a base 64 encoded character string.

Refer to the following program for print document creation.

For the string "Hello World!", to create a print document based on the following settings:

- ❏ Page mode print area (in dots) Origin of horizontal axis: 100, origin of vertical axis: 50, width: 200, height: 100
- ❏ Page mode print positions (in dots) Horizontal print position: 0, vertical print position: 42
- ❏ Font: FontA
- ❏ Scale: x 2 (horizontal) and x 2 (vertical)
- ❏ Style: Bold

```
<script type="text/javascript">
  function createDocument() {
    //Start print document creation.
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                  print">';

    //Create a print document
    //<The page mode starts>
    request += '<page>';

    //<Specify the page mode print area>
    request += '<area x="100" y="50" width="200" height="100"/>';

    //<Specify the page mode print position>
    request += '<position x="0" y="42"/>';

    //<Set print text>
    request += '<text lang="en"/>';
    request += '<text smooth="true"/>';
    request += '<text font="font_a"/>';
    request += '<text width="2" height="2"/>';
    request += '<text em="true"/>';

    //<Specify the character string to print>
    request += '<text>Hello,</text>';
    request += '<position x="0" y="96"/>';
    request += '<text>World!</text>';

    //<The page mode ends>
    request += '</page>';

    //<Specify the feed cut>
    request += '<cut type="feed"/>';

    //End print document creation.
    request += '</epos-print>';

  }
</script>
```

Print image

Transmission of Print Document

A print document is sent using an ePOS-Print object.

Create an ePOS-Print object using the constructor and specify the end point address for the printer to be used for printing as well as the print document into the send method to send the document.

For the details about the printer end point address, refer to "Printer End Point Address" on page 122.

Refer to the following program.

NOTE	TM-S9000II-NW checks the status of the TM-S9000II-NW used for printing and then start printing operation.

To Call SOAP/HTTP Interface

Refer to the following program.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>TITLE</title>
  <script type="text/javascript">
    function sendDocument() {
      //Create a print document
      let request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                    print">';
      request += '<text lang="en" smooth="true"/>';
      request += '<text font="font_a"/>';
      request += '<text width="3" height="3">Hello, World!&#10;</text>';
      request += '<cut type="feed"/>';
      request += '</epos-print>';

      //Create a SOAP envelop
      const soap = '<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">' +
                   '<s:Body>' + request + '</s:Body></s:Envelope>';

      //Set the end point address
      const url = 'http://192.168.192.168/cgi-bin/epos/service.cgi?timeout=10000';

      //<Header settings>
      const options = {
        method: 'POST',
        headers: {
          'Content-Type': 'text/xml; charset=utf-8',
          'If-Modified-Since': 'Thu, 01 Jan 1970 00:00:00 GMT',
          'SOAPAction': '""',
        },
        body: soap,
      };

      // Send print document
      fetch(url, options);
    }

  </script>
</head>
<body>
</body>
</html>
```

Printer End Point Address

Specify the printer end point address in the following format:

To Call "SOAP/HTTP Interface" on page 112

http://[IP address]/cgi-bin/epos/service.cgi?timeout=[timeout time]

Items to specify	Description
IP address	Specify either the IP address or the domain name of TM-S9000II-NW.
Timeout period	Specifies the time to abort the process in milliseconds. The maximum value is 300 seconds (300000). The timeout parameter is optional; when it is omitted, 60 seconds (60000) is set. When the timeout period elapses, the print job is canceled; the data already interpreted by the printer before the start of the print abort process is printed.

Reception of Print Result

Set a callback function using the response root element (p.132) to receive print results.

The following information is obtained:

- ❏ Print result
- ❏ Error code
- ❏ Printer status
- ❏ Printer's battery status

NOTE	The printer status can be obtained when communication with the printer is possible.

Refer to the following program. For the details about how to program a callback function in detail, refer to "Error handling" on page 125.

```
<!DOCTYPE html>
<html>

<head>
   <meta charset="utf-8" />
   <title>TITLE</title>
   <script type="text/javascript">
      function sendDocument() {
         //Create a print document
         let request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                        print">';
         request += '<text lang="en" smooth="true"/>';
         request += '<text font="font_a"/>';
         request += '<text width="3" height="3">Hello, World!&#10;</text>';
         request += '<cut type="feed"/>';
         request += '</epos-print>';

         //Create a SOAP envelop
         const soap = '<?xml version="1.0" encoding="utf-8"?>' +
                      '<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
                       <s:Body>' + request + '</s:Body></s:Envelope>';

         //Set the end point address
         const url = 'http://192.168.192.168/cgi-bin/epos/service.cgi?timeout=10000';

         //<Header settings>
         const options = {
            method: 'POST',
            headers: {
               'Content-Type': 'text/xml; charset=utf-8',
               'If-Modified-Since': 'Thu, 01 Jan 1970 00:00:00 GMT',
               'SOAPAction': '""',
            },
            body: soap,
         };
```

```
        // Send print document
        fetch(url, options)
          .then((response) => {
            if (response.status == 200) {
              response.text().then((resXML) => {
                //Obtain the response root element
                const parser = new DOMParser();
                const res = parser.parseFromString(resXML, 'text/xml');
                const success = res.getElementsByTagName('response')[0].
                                    getAttribute('success');

                //When the printing is not successful, display a message
                if (!/^(1|true)$/.test(success)) {
                  alert('A print error occurred');
                }
              });
            }
          });
      }
    </script>
</head>
<body>
</body>
</html>
```

Error handling

Refer to the following program for the error handling method by a callback function.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>TITLE</title>
  <script type="text/javascript">
    function sendDocument() {
      //Create a print document
      let request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-
                    print">';
      request += '<text lang="en" smooth="true"/>';
      request += '<text font="font_a"/>';
      request += '<text width="3" height="3">Hello, World!&#10;</text>';
      request += '<cut type="feed"/>';
      request += '</epos-print>';

      //Create a SOAP envelop
      const soap = '<?xml version="1.0" encoding="utf-8"?>' +
                   '<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
                   <s:Body>' + request + '</s:Body></s:Envelope>';

      //Set the end point address
      const url = 'http://192.168.192.168/cgi-bin/epos/service.cgi?timeout=10000';

      //<Header settings>
      const options = {
        method: 'POST',
        headers: {
          'Content-Type': 'text/xml; charset=utf-8',
          'If-Modified-Since': 'Thu, 01 Jan 1970 00:00:00 GMT',
          'SOAPAction': '""',
        },
        body: soap,
      };

      // Send print document
      fetch(url, options)
        .then((response) => {
          if (response.status == 200) {

            response.text().then((resXML) => {
              // Obtain the print result and error code
              const parser = new DOMParser();
              const res = parser.parseFromString(resXML, 'text/xml').
                          getElementsByTagName('response');
              let msg = 'Print' + (res[0].getAttribute('success') ? 'Success' : 'Failure') +
                        '\nCode:' + res[0].getAttribute('code') + '\nStatus:\n';

              // Obtain the printer status
              const asb = res[0].getAttribute('status');
              if (asb & 0x00000001) {
                msg += ' No printer response\n';
              }
              if (asb & 0x00000002) {
                msg += ' Print complete\n';
              }
              if (asb & 0x00000004) {
                msg += ' Status of the drawer kick number 3 connector pin = "H"\n';
              }
              if (asb & 0x00000008) {
                msg += ' Offline status\n';
              }
```

```
                    if (asb & 0x00000020) {
                      msg += ' Cover is open\n';
                    }
                    if (asb & 0x00000040) {
                      msg += ' Paper feed switch is feeding paper\n';
                    }
                    if (asb & 0x00000100) {
                      msg += ' Waiting for online recovery\n';
                    }
                    if (asb & 0x00000200) {
                      msg += ' Panel switch is ON\n';
                    }
                    if (asb & 0x00000400) {
                      msg += ' Mechanical error generated\n';
                    }
                    if (asb & 0x00000800) {
                      msg += ' Auto cutter error generated\n';
                    }
                    if (asb & 0x00002000) {
                      msg += ' Unrecoverable error generated\n';
                    }
                    if (asb & 0x00004000) {
                      msg += ' Auto recovery error generated\n';
                    }
                    if (asb & 0x00020000) {
                      msg += ' No paper in the roll paper near end detector\n';
                    }
                    if (asb & 0x00080000) {
                      msg += ' No paper in the roll paper end detector\n';
                    }
                    //Display in the dialog box
                    alert(msg);
                  });
                }
              });
        }
    </script>
</head>
<body>
</body>
</html>
```

Checking the Printer Status

To check the printer status without printing, send empty print data.

Refer to the following program.

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>TITLE</title>
  <script type="text/javascript">
    function sendDocument() {

      //Set the end point address
      const url = 'http://192.168.192.168/cgi-bin/epos/service.cgi?timeout=10000';

      //Create an empty print document to check the printer status
      const soap = '<?xml version="1.0" encoding="utf-8"?>' +
        '<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"><s:Body>' +
        '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"/>' +
        '</s:Body></s:Envelope>';

      //<Header settings>
      const options = {
        method: 'POST',
        headers: {
          'Content-Type': 'text/xml; charset=utf-8',
          'If-Modified-Since': 'Thu, 01 Jan 1970 00:00:00 GMT',
          'SOAPAction': "",
        },
        body: soap,
      };

      //Send empty print data
      fetch(url, options)
        .then((response) => {
          if (response.status == 200) {

            response.text().then((resXML) => {
              //Obtain the response root element
              const parser = new DOMParser();
              const res = parser.parseFromString(resXML, 'text/xml');
              const success = res.getElementsByTagName('response')[0].
                                  getAttribute('success');
              if (/^(1|true)$/.test(success)) {
                alert('Success');
              }
            });
          }
        });
    }
  </script>
</head>
<body>
</body>
</html>
```

ePOS-Print XML Reference

This chapter describes available commands of ePOS-Print XML.

ePOS-Print XML

ePOS-Print XML offers the following commands (elements).

XML for Controlling Printer

Route element	Element	Function
<epos-print>	<text>	Prints a text
	<feed>	Feeds paper
	<image>	Prints a raster image
	<logo>	Prints an NV logo
	<barcode>	Prints a barcode
	<symbol>	Prints a 2D symbol
	<page>	Page mode
	<area>	Specifies a print area for the page mode
	<direction>	Specifies the printing direction for the page mode
	<position>	Specifies a position to start printing for the page mode
	<cut>	Cuts paper
	<pulse>	Drawer kick-out
	<command>	Inserts a command
	<recovery>	Recovers from an error
	<reset>	Resets the printer
<response>		Acquires the execution result.

The print mode includes the standard mode and page mode.

- Standard mode

  Prints data line by line. Line spacing is automatically adjusted according to character size, image size, and barcode height. Suitable for printing receipts whose print length varies by print content.

- Page mode

  Prints page by page. Prints characters, images, and/or barcodes on a single page that is defined as a print area.

  To print in the page mode, switch to the page mode using <page>.

The elements that can be used in each mode are described in the table below.

Route element	Element	Standard mode	Page mode
<epos-print>	<text>	✓	✓
	<feed>	✓	✓
	<image>	✓	✓
	<logo>	✓	✓
	<barcode>	✓	✓
	<symbol>	✓	✓
	<page>	✓	-
	<area>	-	✓
	<direction>	-	✓
	<position>	-	✓
	<cut>	✓	-
	<pulse>	✓	-
	<command>	✓	✓
	<recovery>	✓	-
	<reset>	✓	-
<response>		✓	✓

The table below shows the default for each of the elements.

The default is a factory default configured on the printer.

Element	Attribute	Default value
<text>	lang	"en"
	font	"font_a"
	smooth	"false" / "0"
	dw	"false" / "0"
	dh	"false" / "0"
	width	"1"
	height	"1"
	reverse	"false" / "0"
	ul	"false" / "0"
	em	"false" / "0"
	color	"color_1"
	x	"0"
	y	"21"
	align	"left"
	rotate	"false" / "0"
	linespc	For details, refer to Printer-specific Support Information.
<feed>	linespc	
<image>	color	"color_1"
	align	"left"
	mode	"mono"
<logo>	align	"left"
<barcode>	hri	"none"
	font	"font_a"
	width	"3"
	height	"162"
	align	"left"
	rotate	"false" / "0"
<symbol>	level	The value differs by type of 2D symbol. For details, refer to <symbol>.
	width	
	hight	
	size	
	align	"left"
	rotate	"false" / "0"
<area>	x	"0"
	y	"0"
	width	For details, refer to Printer-specific Support Information..
	height	
<direction>	dir	"left_to_right"
<position>	x	"0"
	y	"21"

<epos-print>

This is an XML document that is sent from an application to a printer.

It requests the printer to execute the specified function. The <epos-print> provides all elements necessary to control a printer.

Attribute

force

Enters in a forced transmission mode.

In the forced transmission mode, print commands are forcibly transmitted to the printer.

Attribute value	Description
"true" / "1"	Enters in the forced transmission mode
"false" / "0"	Enters in a normal transmission mode

Sample program

❏ Blank print document

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print" />
```

❏ Kicks the drawer out while the printer is in an offline status (paper end)

```
<epos-print xmlns= "http://www.epson-pos.com/schemas/2011/03/epos-print"
force="true">
   <pulse />
</epos-print>
```

Supplementary explanation

❏ Use the forced transmission mode while the printer is offline.

Sending it while the printer is online will result in an error.

❏ The following functions are available in the forced transmission mode:

Function	Element
Opens a cash drawer	<pulse>
Recovers from a recoverable error	<recovery>
Reset	<reset>
Sends a real time command	<command>

<response>

This is an XML document that is sent back from a printer to an application.

Attribute

Attribute	Description
success	Print result
code	Error code
status	Status
battery	Battery status

success

Attribute value	Description
"true" / "1"	Print succeeded
"false" / "0"	Failed to print

code

Attribute value	Description
"EPTR_AUTOMATICAL"	Automatic recovery error occurred
"EPTR_COVER_OPEN"	Cover open error occurred
"EPTR_CUTTER"	Auto cutter error occurred
"EPTR_MECHANICAL"	Mechanical error occurred
"EPTR_REC_EMPTY"	No paper is left in the roll paper end detector
"EPTR_UNRECOVERABLE"	Unrecoverable error occurred
"SchemaError"	Error exists in the requested document syntax
"PrintSystemError"	Error occurred with the printing system
"EX_BADPORT"	An error occurred with the communication port
"EX_TIMEOUT"	Print timeout occurred
"ERROR_WAIT_EJECT"	Waiting for paper removal.

status

One or more of the following values that represent the printer status are combined using bitwise OR operator, and expressed in decimal notation.

Attribute value	Description
"0x00000001"	No response from TM-S9000II-NW
"0x00000002"	Printing completed
"0x00000004"	Drawer kick connector pin No.3 status = "H"
"0x00000008"	Offline status
"0x00000020"	Cover is open
"0x00000040"	Paper is being fed by the paper feed switch
"0x00000200"	Paper feed switch is held depressed
"0x00000400"	Mechanical error occurred
"0x00000800"	Auto cutter error occurred

Attribute value	Description
"0x00002000"	Unrecoverable error occurred
"0x00004000"	Automatic recovery error occurred
"0x00010000"	Waiting for insertion of a slip sheet for slip printing
"0x00020000"	Roll paper has almost run out
"0x00040000"	Waiting for ejection of a slip sheet for slip printing
"0x00080000"	Roll paper has run out

battery

0 is always returned.

Sample program

❏ When the printer runs out of paper during printing and failed to print

```
<response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
success="false" code="EPTR_REC_EMPTY" status="252641308" battery="0" />
```

Supplementary explanation

❏ Timing of the printer response differs by printer status.

Printer status	Description
The power is on	The timing set on the printer
The power is off	The timing when a client timeout occurs
Printing: an error occurs	Before printing
Printing: print succeeded	After printing completed

<text>

This element allows you to specify text string to print and print settings such as character style, print position, line spacing.

Attribute

lang

Attribute value	Description
"en"	English (ANK specification)

font

Attribute value	Description
"font_a"	Font A
"font_b"	Font B

smooth

Attribute value	Description
"true" / "1"	Enables smoothing.
"false" / "0"	Disables smoothing.

dw

Attribute value	Description
"true" / "1"	Enables double width size.
"false" / "0"	Disables double width size.

dh

Attribute value	Description
"true" / "1"	Enables double height size.
"false" / "0"	Disables double height size.

width

Attribute value	Description
"1" to "8"	Specifies the horizontal scaling factor rate.

height

Attribute value	Description
"1" to "8"	Specifies the vertical scaling factor rate.

reverse

Attribute value	Description
"true" / "1"	Inverts black and white
"false" / "0"	Cancels the black and white inversion

ul

Attribute value	Description
"true" / "1"	Enables the underscore style.
"false" / "0"	Disables the underscore style.

em

Attribute value	Description
"true" / "1"	Enables emphasized style.
"false" / "0"	Disables emphasized style.

color

Attribute value	Description
"none"	No printing
"color_1"	First color

x

Attribute value	Description
"0" to "65535"	Specifies the horizontal print start position in units of dots.

y

Attribute value	Description
"0" to "65535"	Specify the vertical print start position in units of dots.

align

Attribute value	Description
"left"	Left alignment
"center"	Center alignment
"right"	Right alignment

rotate

Attribute value	Description
"true" / "1"	Enables text rotation.
"false" / "0"	Disables text rotation.

linespc

Attribute value	Description
"0" to "255"	Specifies paper feed amount per line in units of dots.

Sample program

When printing a text string with the following settings

Item	Value
Language	English
Smoothing	Enabled
Alignment	Center alignment
Font	Font B
Double size	Double height and width size
Underscore	Enabled

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <text lang="en" />
   <text smooth="true" />
   <text align="center" />
   <text font="font_a" />
   <text dw="true" dh="true" />
   <text ul="true" />
   <text>Hello, World!&#10;</text>
</epos-print>
```

Supplementary explanation

- ❏ To print data other than text after printing text, start a new line or feed paper.
- ❏ In the page mode, text is printed from the current print position with the base line dot of the characters as the standard. For information on the base line dot, refer to Printer-specific Support Information.
- ❏ For the character codes available for printing, refer to the Technical Reference Guide.
- ❏ If "dw" and "width" are set in a single element, the "width" setting is applied.
- ❏ If "dh" and "length" are set in a single element, the "height" setting is applied.
- ❏ Use "y" in the page mode.
- ❏ The "align" and "rotate" cannot be applied in the page mode.
- ❏ To rotate text string in the page mode, use <direction>.
- ❏ When setting "align" or "rotate", set it at the beginning of a line.
- ❏ The "align" setting specified in this element is also applied to <image>, <logo>, <barcode>, and <symbol>.
- ❏ The "rotate" setting specified in this element is also applied to <barcode> and <symbol>.
- ❏ When specifying the lang attribute, specify it at the beginning of the print request.

<feed>

This element allows you to specify paper feed amount in dots or by the number of lines, line spacing, and paper position after printing a line.

If you do not specify the paper feed amount, paper is advanced one line.

Attribute

unit

Attribute value	Description
"0" to "255"	Specifies paper feed amount in units of dots

line

Attribute value	Description
"0" to "255"	Specifies paper feed amount in lines

linespc

Attribute value	Description
"0" to "255"	Specifies paper feed amount per line in units of dots

pos

Attribute value	Description
"peeling"	Feed the sheet to the peeling position
"cutting"	Feed the sheet to the cut position
"current_tof"	Feed the sheet to the top of the current label
"next_tof"	Feed the sheet to the top of the next label

Sample program

❏ Advances paper three lines

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <feed line="3" />
</epos-print>
```

❏ Inserts a line break after printing text string

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <text>Hello</text><feed />
   <text>World</text><feed />
</epos-print>
```

Supplementary explanation

- ❏ The "pos" setting is ignored in the page mode.
- ❏ When "30" dots is specified in "linespc", up to 240 lines can be specified in "line".
- ❏ The line spacing setting is stored for each of the standard mode and the page mode. The "linespc" setting specified in this element affects the linespc setting of <text> and <feed>, which may follow.
- ❏ If the line spacing for a single line is set smaller than the print character size, paper may be fed for a larger quantity than the set amount to ensure proper printing.

<image>

This element allows you to specify a raster image. (data type xs: base64Binary)

Raster images are composed of pixels arranged in a grid formation. The top left corner is the origin.

Two colors (black and white) uses 1 bit per pixel, and 16 colors uses 4 bits per pixel. The bits are arranged from the highest bit of each byte. Zero padding is needed so that each line of the image is expressed in byte units.

Attribute

NOTE	• "width" and "height" must be specified. • If "color" and "mode" are not specified, the values shown below are set. ∗ **color: color_1** ∗ **mode: mono**

width

Attribute value	Description
"0" to "655635"	Specifies the image width in units of dots

height

Attribute value	Description
"0" to "655635"	Specifies the image height in units of dots

color

Attribute value	Description
"none"	No printing
"color_1"	First color

align

Attribute value	Description
"left"	Left alignment
"center"	Center alignment
"right"	Right alignment

mode

Attribute value	Description
"mono"	Black and white (2 colors)
"gray16"	Grayscale (16-step)

Sample program

Prints a 8 dots width and 8dots height raster image that is filled with solid color.

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <image width="8" height="8">//////////8=</image>
</epos-print>
```

Supplementary explanation

- ❏ Grayscale printing is allowed in the standard mode, not allowed in the page mode.
- ❏ We cannot guarantee the accuracy of reading a barcode or 2D symbol printed in grayscale. Print it in black and white.
- ❏ In order to print a raster image at a high speed, set "align" to "left" and set "width" to a multiple of 8 which does not exceed the sheet width of the printer.
- ❏ In the page mode, set the print position of the image so that the image does not extend beyond the print area.
- ❏ If you set to print a raster image in grayscale, the amount of data of the image increases and may be printed intermittently, which causes white streaks on printout.
- ❏ In the page mode, the bottom left corner of the raster image is aligned with the start position for printing the image. The print start position is not automatically moved.
- ❏ The "align" setting is ignored in the page mode.
- ❏ When setting "align" in the standard mode, set it at the beginning of a line.
- ❏ The "align" setting specified in this element is also applied to <text>, <logo>, <barcode>, and <symbol>.
- ❏ Create a raster image using the ePOS-Print XML generation tool or your application. When using your application, follow the instruction below for 2 colors image or 16 colors image, whichever you want to create.

Attribute value	Description
Black and white	Set the image width to a multiple of 8, or pad the lacked bits with zeros.
16-step grayscale	Set the image width to a multiple of 2, or pad the lacked bits with zeros.

- ❏ The composition of the image data in raster format is as follows.

When the image is in two colors (black and white)

White: 0, Black: 1

Example:

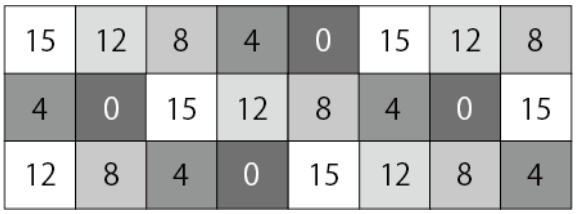
1	0	1	0	0	0	1	1
0	1	1	1	0	1	1	1
0	0	0	1	1	0	0	0

Step	Description
1	Convert the image to Base64 (byte array) With the upper left as the origin, and with one bit per pixel, pad the data with "0" so that each line data is in bytes. 0xA3 0x77 0x18
2	Encode the image to Base64 o3cy
3	Description <image width="8" height="3" color="color_1" mode="mono">o3cY</image>

When the image is in multiple tones (16 shades)

White: 15 or larger, Black: 0

Example:



Step	Description
1	Convert the image to Base64 (byte array) With the upper left as the origin, and with four bits per pixel, pad the data with "0" so that each line data is in bytes. 0xFC 0x84 0x0F 0xC8 0x40 0xFC 0x84 0x0F 0xC8 0x40 0xFC 0x84
2	Encode the image to Base64 /IQPyED8hA/IQPyE
3	Description <image width="8" height="3" color="color_1" mode="gray16">/IQPyED8hA/IQPyE</image>

<logo>

This element allows you to print an NV logo registered in the NV memory of the printer.

The NV logo needs to be registered to the printer in advance. For how to register the NV logo, refer to the Technical Reference Guide.

Attribute

NOTE	"key1" and "key2" must be specified.

key1

Attribute value	Description
"0" to "255"	Specifies the key code 1 of the NV logo

key2

Attribute value	Description
"0" to "255"	Specifies the key code 2 of the NV logo

align

Attribute value	Description
"left"	Left alignment
"center"	Center alignment
"right"	Right alignment

Sample program

Prints an NV logo whose key code 1 and 2 are stored in 48.

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <logo key1="48" key2="48" />
</epos-print>
```

Supplementary explanation

- ❏ Grayscale printing is allowed in the standard mode, not allowed in the page mode.
- ❏ In the page mode, the bottom left corner of the logo is aligned with the start position for printing the logo.
- ❏ The "align" setting is ignored in the page mode.
- ❏ When setting "align" in the standard mode, set it at the beginning of a line.
- ❏ The "align" setting specified in this element is also applied to <text>, <logo>, <barcode>, and <symbol>.

<barcode>

Specifies barcode data as a text string.

Barcode type

Type	Description
UPC-A	If an 11-digit figure is specified, the check digit is automatically appended. If a 12-digit figure is specified, the 12th digit is used as the check digit but verification is not performed.
UPC-E	Specify 0 in the first digit. Specify the manufacturer code in the 2nd to 6th digits. Specify the item code in right justification in the 7th to 11th digits. The number of digits of the item code depends on the manufacturer code. Specify 0 in each unused data. If an 11-digit figure is specified, the check digit is automatically appended. If a 12-digit figure is specified, the 12th digit is used as the check digit but verification is not performed.
EAN13 JAN13	If an 12-digit figure is specified, the check digit is automatically appended. If a 13-digit figure is specified, the 13th digit is used as the check digit but verification is not performed.
EAN8 JAN8	If an 7-digit figure is specified, the check digit is automatically appended. If an 8-digit figure is specified, the 8th digit is used as the check digit but verification is not performed.
CODE39	If the first character is * , this character is processed as the start character. Otherwise, the start character is automatically added.
ITF	The start and stop codes are automatically added. Addition and verification of the check digit are not performed.
CODABAR	Specify the start character ((A to D, a to d). Specify the stop character ((A to D, a to d). Addition and verification of the check digit are not performed.
CODE93	The start and stop characters are automatically added. The check digit is automatically calculated and added.
CODE128	Specify the start character (CODE A, CODE B, CODE C). The stop character is automatically added. The check digit is automatically calculated and added. To encode the following characters, specify the corresponding 2-digit code starting with { : • FNC1: {1 • FNC2: {2 • FNC3: {3 • FNC4: {4 • CODE A: {A • CODE B: {B • CODE C: {C • SHIFT: {S • {: {{ When specifying CODE C, specify the barcode data as the control code of the escape sequence.

Type	Description
GS1-128	The start character, FNC1, check digit, and stop characters are automatically added. To automatically calculate and add the application ID (AI) and the following check digit, specify "*" at the check digit position. The application ID (AI) can be put in parentheses. The parentheses are used as print characters for HRI and not encoded as data. A blank space can be inserted between the application ID (AI) and data. The blank space is used as print characters for HRI and not encoded as data. To encode the following characters, specify the corresponding 2-digit code starting with { : • FNC1{1 • FNC3{3 • ({(•){) • *{* • {{{
GS1 DataBar Omnidirectional, GS1 DataBar Truncated GS1 DataBar Limited	Specify a 13-digit product ID (GTIN) excluding the application ID (AI) and check digit.
GS1 DataBar Expanded	The application ID (AI) can be put in parentheses. The parentheses are used as print characters for HRI and not encoded as data. To encode the following characters, specify the corresponding 2-digit code starting with { : • FNC1: {1 • (: {(•): {)

When specifying binary data which cannot be represented as a text string, use the following escape sequences.

Text string	Description
\xnn	Control code (set nn in hexadecimal)
\\	Back slash

Attribute

NOTE	"type" must be specified.

type

Attribute value	Description
"upc_a"	UPC-A
"upc_e"	UPC-E
"ean13"	EAN13
"jan13"	JAN13
"ean8"	EAN8
"jan8"	JAN8
"code39"	CODE39
"itf"	ITF
"codabar"	CODABAR
"code93"	CODE93
"code128"	CODE128
"gs1_128"	GS1-128
"gs1_databar_omnidirectional"	GS1 DataBar Omnidirectional
"gs1_databar_truncated"	GS1 DataBar Truncated
"gs1_databar_limited"	GS1 DataBar Limited
"gs1_databar_expanded"	GS1 Databar Expanded

hri

Attribute value	Description
"none"	No printing.
"above"	Above the barcode
"below"	Below the barcode
"both"	Both above and below the barcode

font

Attribute value	Description
"font_a"	Font A
"font_b"	Font B

width

Attribute value	Description
"2" to "6"	Specifies the width of a single module in units of dots

height

Attribute value	Description
"1" to "255"	Specifies the height of a single module in units of dots

align

Attribute value	Description
"left"	Left alignment
"center"	Center alignment
"right"	Right alignment

rotate

Attribute value	Description
"true" / "1"	Enables text rotation
"false" / "0"	Disables text rotation

Sample program

Prints multiple types of barcode

```
epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <barcode type="upc_a" width="2" height="64" hri="below">01234567890</barcode>
   <barcode type="upc_e">01234500005</barcode>
   <barcode type="ean13">201234567890</barcode>
   <barcode type="jan13">201234567890</barcode>
   <barcode type="ean8">2012345</barcode>
   <barcode type="jan8">2012345</barcode>
   <barcode type="code39">ABCDE</barcode>
   <barcode type="itf">012345</barcode>
   <barcode type="codabar">A012345A</barcode>
   <barcode type="code93">ABCDE</barcode>
   <barcode type="code128">{Babcde</barcode>
   <barcode type="gs1_128">(01)201234567890*</barcode>
   <barcode type="gs1_databar_omnidirectional">0201234567890</barcode>
   <barcode type="gs1_databar_truncated">0201234567890</barcode>
   <barcode type="gs1_databar_limited">0201234567890</barcode>
   <barcode type="gs1_databar_expanded">(01)2012345678903</barcode>
</epos-print>
```

Supplementary explanation

- ❏ When Standard mode is selected, this command is enabled only when the print position is at the head of a line or when no data exists in the print buffer.
- ❏ In the page mode, the bottom left corner of the barcode (except HRI) is aligned with the start position for printing the barcode.
- ❏ If your barcode settings are not compliant with the barcode standards, or if the size is larger than the print area, the barcode is not printed with no error returned.
- ❏ When setting "align" or "rotate" in the standard mode, set it at the beginning of a line.
- ❏ The "align" and "rotate" settings are ignored in the page mode.
- ❏ To rotate text string in the page mode, use <direction>.
- ❏ The "align" setting specified in this element is also applied to <text>, <image>, <logo>, and <symbol>.
- ❏ The "rotate" setting specified in this element is also applied to <text> and <symbol>.

<symbol>

This element allows you to specify a 2D symbol using a text string.

2D symbol type

Type	Description
PDF417	Converts the string into UTF-8, processes the escape sequence(s), and encodes the data. The maximum number of code words in the data area is 928, the maximum number of code words in a single stage is 30, and the maximum number of stages is 90.
QR Code	Converts the string into JIS, processes the escape sequence(s), and encodes the data by choosing the data type from the following: • Numeric data: 0 to 9 • Alphanumeric data: 0 to 9, A to Z, space, $, %, *, +, -, ., /, : • Kanji character: Shift JIS code • 8-bit byte data: 0X00 to 0xff
MaxiCode	Converts the string into UTF-8, processes the escape sequence(s), and encodes the data. In Mode 2 or 3, if the first data is []>\x1e01\x1dyy ("yy" is a 2-digit figure), this is processed as the message header and the second and succeeding data sequence is processed as the primary message. Otherwise, the primary message starts with the first data. Specify the primary message in the following format. • For Mode 2 Postal code: (1- to 9-digit number) GS: (\x1d) ISO country code: (1- to 3-digit number) GS: (\x1d) Service class code (1- to 3-digit number) • For Mode 3 Postal code: (data which can be converted with 1 to 6 code sets A) GS: (\x1d) ISO country code: (1- to 3-digit number) GS: (\x1d) Service class code (1- to 3-digit number)
GS1 DataBar Stacked GS1 DataBar Stacked Omnidirectional	Converts the string into UTF-8, processes the escape sequence(s), and encodes the data. Specify a 13-digit product ID (GTIN) excluding the application ID (AI) and check digit.
GS1 DataBar Expanded Stacked	Converts the string into UTF-8, processes the escape sequence(s), and encodes the data. The application ID (AI) can be put in parentheses. The parentheses are used as print characters for HRI and not encoded as data. To encode the following characters, specify the corresponding 2-digit code starting with { : • FNC1: {1 • (: {(•): {)

When specifying binary data which cannot be represented as a text string, use the following escape sequences.

Text string	Description
\xnn	Control code (set nn in hexadecimal)
\\	Back slash

Attribute

NOTE	"type" must be specified.

type

Attribute value	Type
"pdf417_standard"	Standard PDF417
"pdf417_truncated"	Truncated PDF417
"qrcode_model_1"	QR Code Model 1
"qrcode_model_2"	QR Code Model 2
"maxicode_mode_2"	MaxiCode Mode 2
"maxicode_mode_3"	MaxiCode Mode 3
"maxicode_mode_4"	MaxiCode Mode 4
"maxicode_mode_5"	MaxiCode Mode 5
"maxicode_mode_6"	MaxiCode Mode 6
"gs1_databar_stacked"	GS1 DataBar Stacked
"gs1_databar_stacked_omnidirectional"	GS1 DataBar Stacked Omnidirectional
"gs1_databar_expanded_stacked"	GS1 DataBar Expanded Stacked

level

❏ PDF417

Attribute value	Description
"level_0"	Error correction level 0
"level_1" (default value)	Error correction level 1
"level_2"	Error correction level 2
"level_3"	Error correction level 3
"level_4"	Error correction level 4
"level_5"	Error correction level 5
"level_6"	Error correction level 6
"level_7"	Error correction level 7
"level_8"	Error correction level 8
"default"	Default value (error correction level 1)

❏ QR Code

Attribute value	Description
"level_l"	Error correction level L
"level_m"	Error correction level M
"level_q"	Error correction level Q
"level_h"	Error correction level H
"default"	Default value (error correction level M)

width

2D symbol type	Valid value	Default value
PDF417	"2" to "8"	"3"
QR Code	"3" to "16"	"3"
MaxiCode	Ignored	
2D GS1 DataBar	"2" to "8"	"2"

height

2D symbol type	Valid value	Default value
PDF417	"2" to "8" (Scaling factor for width)	"3"
QR Code	Ignored	
MaxiCode		
2D GS1 DataBar		

size

2D symbol type		Default value	Description
PDF417		"0" (auto)	Specifies the number of code words per stage.
QR Code		Ignored	
MaxiCode			
2D GS1 DataBar	Expanded Stacked	"0" (auto)	Specifies the maximum width of the 2D symbol (106 or more).
	Other	Ignored	

align

Attribute value	Description
"left"	Left alignment
"center"	Center alignment
"right"	Right alignment

rotate

Attribute value	Description
"true" / "1"	Enables text rotation.
"false" / "0"	Disables text rotation.

Sample program

Prints multiple types of 2D symbols

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <symbol type="pdf417_standard">ABCDE</symbol>
   <symbol type="qrcode_model_2" level="level_q">ABCDE</symbol>
   <symbol type="maxicode_mode_2">908063840\x1d850\x1d001\x1d\x04</symbol>
   <symbol type="gs1_databar_stacked">0201234567890</symbol>
   <symbol type="gs1_databar_stacked_omnidirectional">0201234567890</symbol>
   <symbol type="gs1_databar_expanded_stacked">(01)02012345678903</symbol>
</epos-print>
```

Supplementary explanation

- ❏ If your 2D symbol settings are not compliant with the 2D symbol standards, or if the size is larger than the print area, the symbol is not printed with no error returned.
- ❏ In the standard mode, a 2D symbol whose height exceeds 831 dots cannot be printed.
- ❏ In the page mode, the bottom left corner of the 2D symbol is aligned with the start position for printing the symbol.
- ❏ Specify the "level" that matches the type of the 2D symbol you specified.
- ❏ Set "default" to "level" if the 2D symbol type is MaxiCode or 2D GS1 DataBar.
- ❏ When setting "align" or "rotate" in the standard mode, set it at the beginning of a line.
- ❏ The "align" and "rotate" settings are ignored in the page mode.
- ❏ To rotate text string in the page mode, use <direction>.
- ❏ The "align" setting specified in this element is also applied to <text>, <image>, <logo>, and <barcode>.
- ❏ The "rotate" setting specified in this element is also applied to <text> and <barcode>.

<page>

This element allows you to change to the page mode from the standard mode.

Elements of <page>

Element	Description
<text>	Prints a text
<feed>	Paper feed
<image>	Prints a raster image
<logo>	Prints an NV logo
<barcode>	Prints a barcode
<symbol>	Prints a 2D symbol
<area>	Specifies a print area
<direction>	Specifies the printing direction
<position>	Specifies a position to start printing
<command>	Inserts a command

Sample program

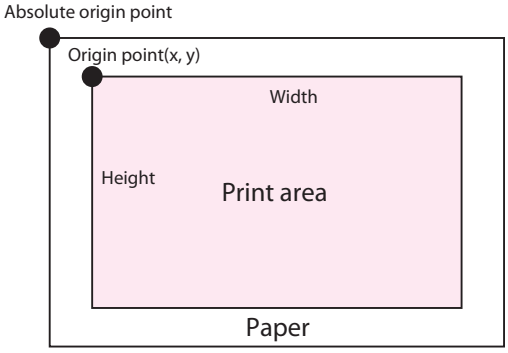
Prints a string "ABCDE" in the page mode

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
    <text>ABCDE</text>
  </page>
</epos-print>
```

<area>

This element allows you to set a print area by specifying an origin (starting position) relative to the absolute origin, and specifying width and height.

The absolute origin is the top-left dot of the printable area.



Attribute

NOTE	The "x", "y", "width" and "height" elements must be specified.

x

Attribute value	Description
"0" to "65535"	Specifies the horizontal origin in units of dots.

y

Attribute value	Description
"0" to "65535"	Specifies the vertical origin in units of dots.

width

Attribute value	Description
"0" to "65535"	Specifies the width in units of dots

height

Attribute value	Description
"0" to "65535"	Specifies the height in units of dots

Sample program

Sets a 200-dots width and 30-dots height print area that starts from x:100 and y:50, and prints a text string "ABCDE".

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
    <area x="100" y="50" width="200" height="30" />
    <text>ABCDE</text>
  </page>
</epos-print>
```

Supplementary explanation

- ❏ This does not work in the standard mode.
- ❏ Define the print area in accordance with the contents to print. Any portion of print data outside the print area is not printed.
- ❏ Specify the width and height of the print area in accordance with the print direction setting. If the width and height of the print area do not match the print direction setting, any portion of print data outside the print area will not be printed. Use <direction> to specify the print direction.

<direction>

This element allows you to specify the print direction in the page mode.

Print direction and rotation of the print area can be specified. The starting position of the print area moves in accordance with the rotation.

Attribute

NOTE	"dir" must be specified.

dir

Attribute value	Description
"left_to_right"	Prints from left to right (no rotation. The starting position is at the upper left.)
"bottom_to_top"	Prints from bottom to top (Rotate the print area counterclockwise by 90 degrees. The starting position is at the lower left.)
"right_to_left"	Prints from right to left (Rotate the print area by 180 degrees. The starting position is at the lower right.)
"top_to_bottom"	Prints from top to bottom (Rotate the print area clockwise by 90 degrees. The starting position is at the upper right.)

Sample program

Rotates the print area clockwise by 90 degrees and prints a text string "ABCDE".

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
    <direction dir="top_to_bottom" />
    <text>ABCDE</text>
  </page>
</epos-print>
```

Supplementary explanation

This does not work in the standard mode.

<position>

This element allows you to specify the print start position (coordinates) within the print area specified by <area> with reference to the starting position of the print area.

Attribute

NOTE	"x" and "y" must be specified.

x

Attribute value	Description
"0" to "65535"	Specifies the horizontal print start position in units of dots.

y

Attribute value	Description
"0" to "65535"	Specify the vertical print start position in units of dots.

Sample program

Specifies the print start position to 50,30 in the print area specified by <area>, and prints a text string "ABCDE".

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
    <area x="100" y="50" width="200" height="100" />
    <position x="50" y="30" />
    <text>ABCDE</text>
  </page>
</epos-print>
```

Supplementary explanation

❏ This does not work in the standard mode.

❏ Define the print start position (coordinates) in accordance with the contents to print.

Print data	Description
Text string	Specify the leftmost position of the baseline for the first character. This can be omitted when printing data with the standard size in left justification. When printing a double-height character, set y to 42 or larger.
Barcode	Specify the lower-left position of the symbol. Specify the height of the barcode in y.
Graphics/logo	Specify the lower-left position of the graphic data. Specify the height of the graphic data in y.
2D symbol	Specify the upper-left position of the symbol. This can be omitted when printing from the upper-left position.

<cut>

This element allows you to cut paper.

Attribute

NOTE	When "type" is not specified, "feed" is applied.

type

Attribute value	Description
"no_feed"	Cut without feed (cut the sheet without feeding paper)
"feed"	Feed cut (cut the sheet after feeding paper)

Sample program

Feeds paper and then cuts it.

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <cut type="feed" />
</epos-print>
```

Supplementary explanation

❏ This does not work in the page mode.

❏ When setting this element, set it at the beginning of a line.

<pulse>

This element allows you to send a signal to the drawer kick connector.

Attribute

NOTE	If "drawer" and "time" are not specified, the values shown below are set. • drawer: drawer_1 • time: pulse_100

drawer

Attribute value	Description
"drawer_1"	Drawer kick connector pin No.2
"drawer_2"	Drawer kick connector pin No.5

time

Attribute value	Description
"pulse_100"	100-msec signal
"pulse_200"	200-msec signal
"pulse_300"	300-msec signal
"pulse_400"	400-msec signal
"pulse_500"	500-msec signal

Sample program

Sends a 100 msec pulse signal to pin No.2 of the drawer kick connector.

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <pulse drawer="drawer_1" time="pulse_100" />
</epos-print>
```

Supplementary explanation

❏ This does not work in the page mode.

<command>

This element allows you to specify ESC/POS commands encoding them in hexadecimal.

Sample program

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <command>41424344450a</command>
</epos-print>
```

Supplementary explanation

- ❏ Refer to the following URL for details of the ESC/POS command.
  https://support.epson.net/publist/reference_en/
- ❏ ePOS-Print Service I/F does not check the commands sent by the <command> XML.
  If the commands interfere with ePOS-Print Service I/F operations, other XML controls may work wrongly or status values may become invalid.
  This XML should be used with a full understanding of ESC/POS commands and the receipt printer specifications.

<recovery>

This element allows you to set the printer to recover from an error.

Sample program

Recovers from an recoverable error and clears the printer buffer.

```
<epos-print xmlns= "http://www.epson-pos.com/schemas/2011/03/epos-print"
force="true">
   <recovery />
</epos-print>
```

Supplementary explanation

- ❏ This does not work in the page mode.
- ❏ Use this element with the forced transmission mode (force) enabled.
- ❏ After recovering from a recoverable error, the buffer of the printer is reset.

<reset>

This element allows you to reset the printer.

Sample program

Resets the printer.

```
<epos-print xmlns= "http://www.epson-pos.com/schemas/2011/03/epos-print">
   <reset />
</epos-print>
```

Supplementary explanation

- ❏ This does not work in the page mode.
- ❏ Any other print commands included in the document are ignored.

TM-S9000II-NW specification

NOTE	Refer to the Technical Reference Guide for more information.

Printer Specifications

Item		Specification
Amount of initial feed		30 dots
Initial page mode area		512 x 831 dots
Maximum page mode area		512 x 1662 dots
Baseline of Font A		21 dots from the top of the character
Baseline of Font B	ANK	16 dots from the top of the character

Elements for which restricted parameter setting values can be specified

Element	Attribute	Specificable Setting Value
<text>	lang	• "en"
	font	• "font_a" • "font_b"
	smooth	• "true" / "1" • "false" / "0"
	dw	• "true" / "1" • "false" / "0"
	dh	• "true" / "1" • "false" / "0"
	width	• "1" to "8"
	height	• "1" to "8"
	reverse	• "true" / "1" • "false" / "0"
	ul	• "true" / "1" • "false" / "0"
	em	• "true" / "1" • "false" / "0"
	color	• "color_1"
	x	• "0" to "65535"
	y	• "0" to "65535"
	align	• "left" • "center" • "right"
	rotate	• "true" / "1" • "false" / "0"
	linespc	• "0" to "255"

Element	Attribute	Specificable Setting Value
<feed>	unit	• "0" to "255"
	line	• "0" to "255"
	linespc	• "0" to "255"
<image>	width	• "0" to "655635"
	height	• "0" to "655635"
	color	• "color_1"
	align	• "left" • "center" • "right"
	mode	• "mono" • "gray16"
<barcode>	type	• "upc_a" • "upc_e" • "ean13" • "jan13" • "ean8" • "jan8" • "code39" • "itf" • "codabar" • "code93" • "code128" • "gs1_128" • "gs1_databar_omnidirectional" • "gs1_databar_truncated" • "gs1_databar_limited" • "gs1_databar_expanded"
	hri	• "none" • "above" • "below" • "both"
	font	• "font_a" • "font_b"
	width	• "2" to "6"
	height	• "1" to "255"
	align	• "left" • "center" • "right"
	rotate	• "true" / "1" • "false" / "0"

Element	Attribute	Specificable Setting Value
<symbol>	type	• "pdf417_standard" • "pdf417_truncated" • "qrcode_model_1" • "qrcode_model_2" • "maxicode_mode_2" • "maxicode_mode_3" • "maxicode_mode_4" • "maxicode_mode_5" • "maxicode_mode_6" • "gs1_databar_stacked" • "gs1_databar_stacked_omnidirectional" • "gs1_databar_expanded_stacked"
	level	• "level_0" • "level_1" (default value) • "level_2" • "level_3" • "level_4" • "level_5" • "level_6" • "level_7" • "level_8" • "level_l" • "level_m" • "level_q" • "level_h" • "default"
	size	• PDF417 • QR Code • MaxiCode • 2D GS1 DataBar
	align	• "left" • "center" • "right"
	rotate	• "true" / "1" • "false" / "0"
<cut>	type	• "no_feed" • "feed"

ePOS-Print Settings

Printer Model	Printing Method	Character Code Tables
TM-S9000II-NW	Thermal (180 dpi)	Page 0-5, 16-19, 20-21, 26,30-31, 11-15, 32-53

Support Tools for Generating ePOS-Print XML Data

ePOS-Print Editor

This section describes how to use ePOS-Print Editor included in the contents in the package.

This tool is a support tool for generating XML data. This tool allows you to create an ePOS-Print XML (p.128) print document as you like. In addition, the generated XML data can be printed for testing. Use this tool for your application development.

ePOS-Print Editor Operating Environment

Web Browser

- Mozilla Firefox 91 or later
- Google Chrome 89 or later
- Microsoft Edge 91 or later
- Safari on iOS 13 or later

* Windows Internet Explorer is not supported.

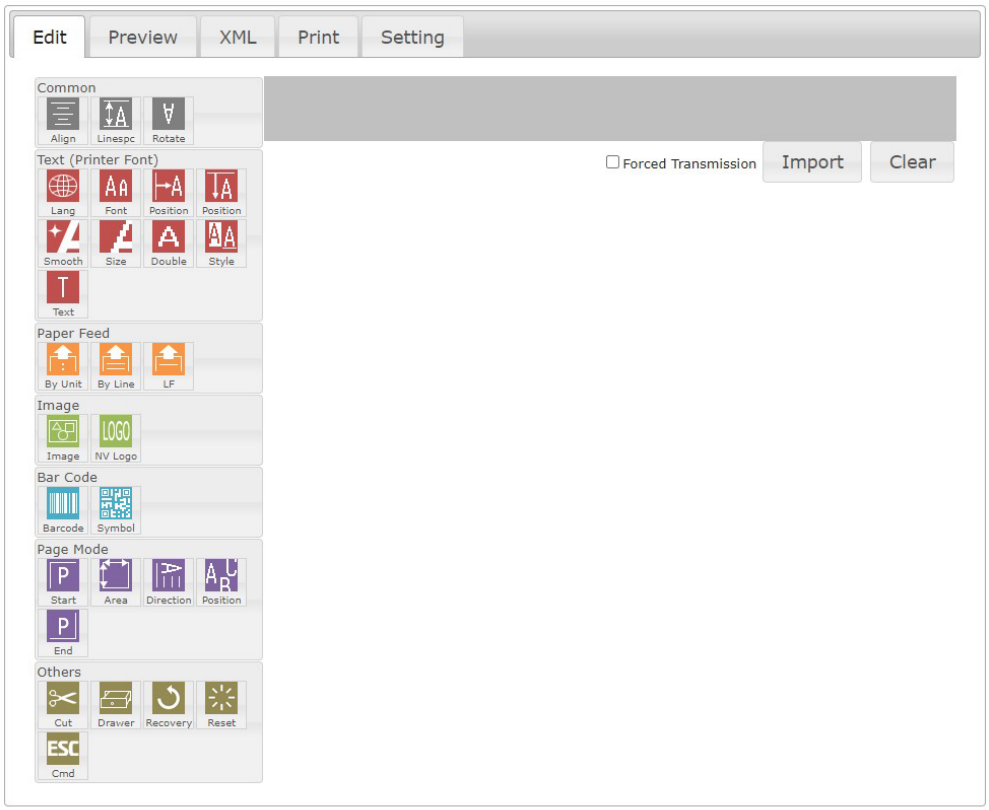
Environment Setting Procedure

NOTE	If opening a page with ePOS-Print Editor's HTML file placed on the local disk, some functionality does not operate due to your Web browser's security policy. Place the HTML file of ePOS Editor to a folder under Web server.

1 **Copy the editor folder contained in the sample program to the Web server.**

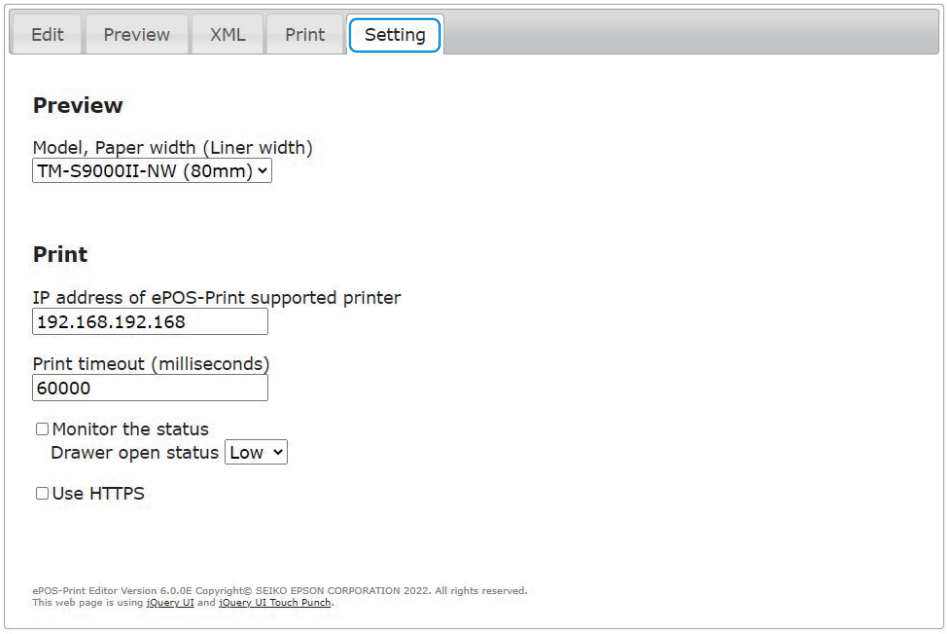
2 **Open the following URL page using the Web browser.**
http://[Web server IP address]/editor/index.html

3 ePOS-Print Editor appears.



Setting

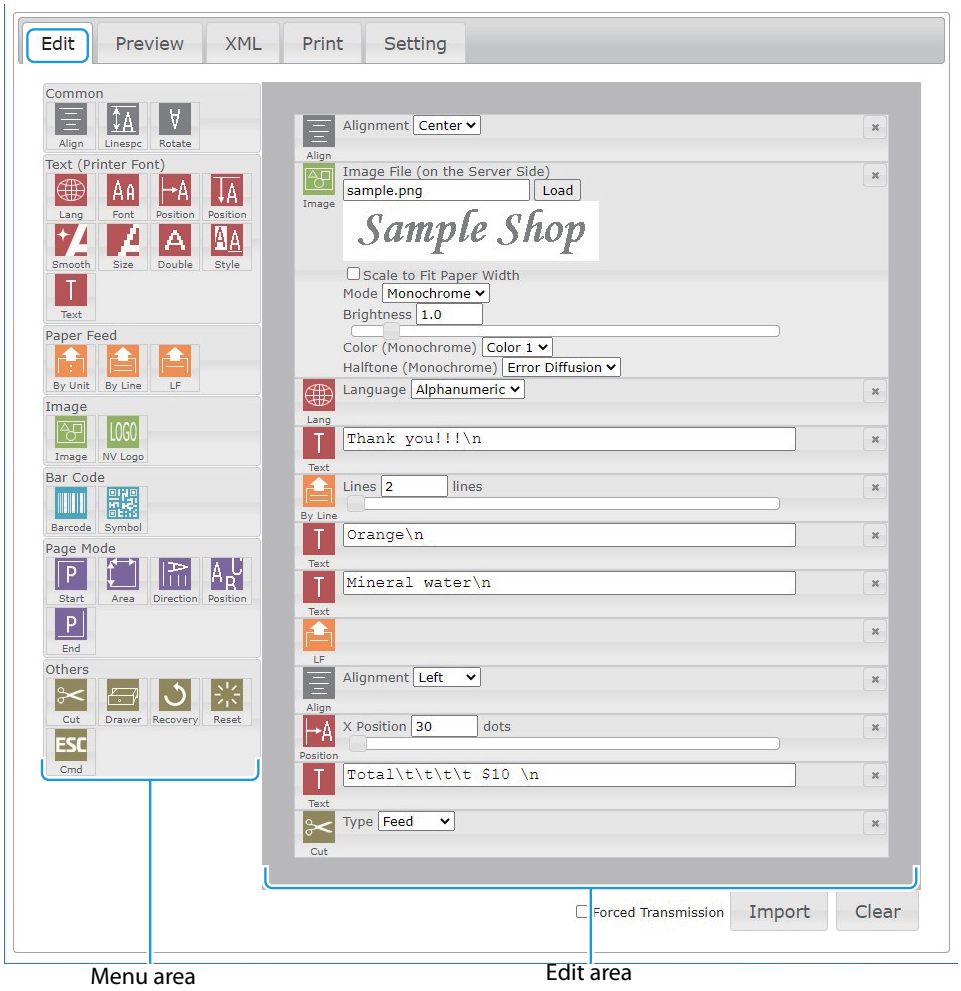
Perform the print setting and the preview setting. Select the [Setting] tab to display the Setting screen.



Item	Description
Model	Specify the series name of the printer to be used for printing. The Preview screen resizes according to the paper width set to the model.
IP address of ePOS-Print supported printer	Specify the IP address of the printer. Be sure to specify this item.
Print timeout (milliseconds)	Specify the print timeout time in milliseconds. The maximum value is 60000 (60 seconds).
Monitor the status	When this checkbox is checked, the printer's status is monitored.
Drawer open status	Change this according to the specifications of the drawer that you are using.
Use HTTPS	Check this to use HTTPS to communicate with the printer to print.

Creating a Sample Code

Select the [Edit] tab to display the Edit screen. Create an ePOS-Print XML print document in the Edit screen.



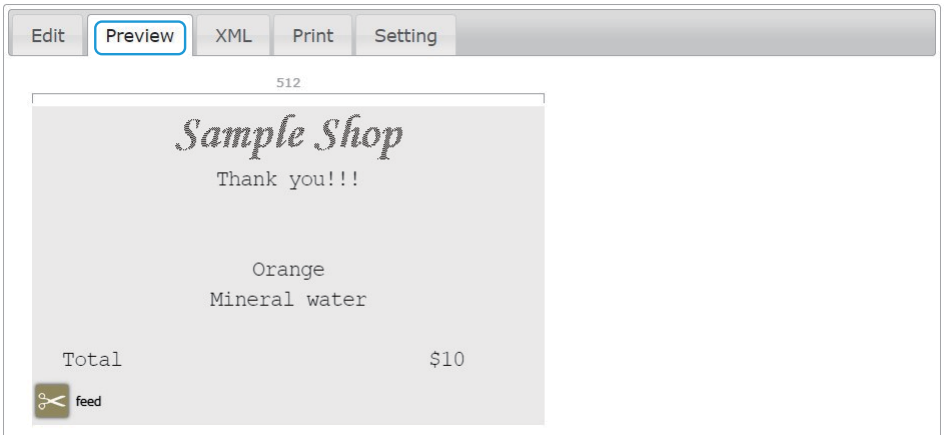
Item	Description
Menu area	Displays the available functions. Click an icon to add it to the bottom of the edit area, and drag an icon to insert it anywhere in the edit area.
Edit area	Displays the functions selected in the menu area. Drag an element to change its position. An element can be deleted using the x button located on its right side.
Import	ePOS-Print Editor can import an ePOS-Print XML print document. For details, refer to "Import" on page 171.
Clear	Deletes the edited details.
Forced Transmission	Sets forced transmission mode.

Create XML data as follows:

1 **Click an icon in the menu area to add an element in the edit area.**
The position of the added function can be changed by dragging.

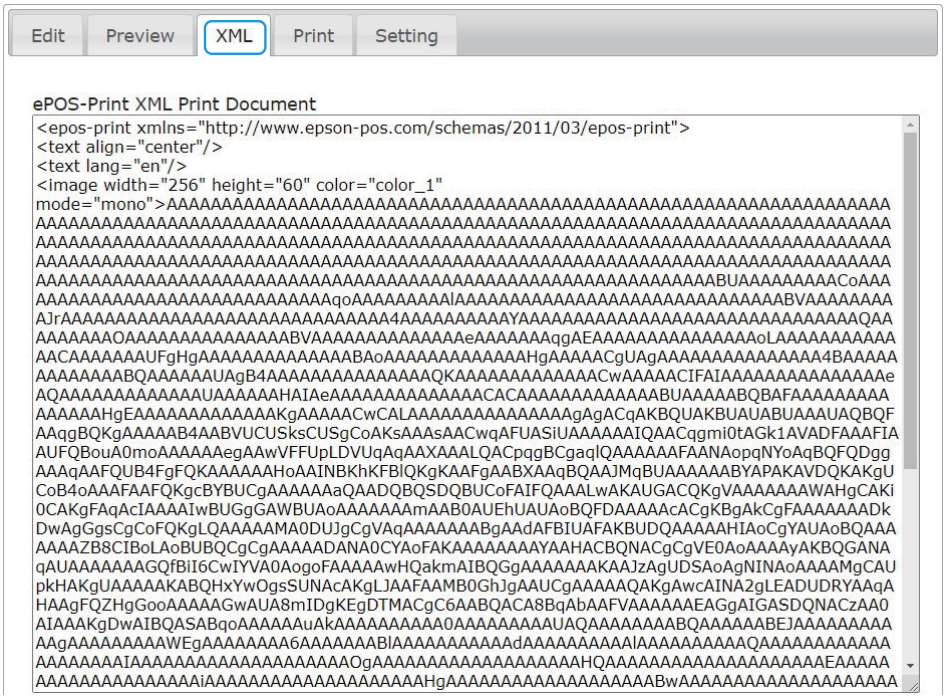
2 **Configure the added element.**
Example: When the NV logo is added, set the key code.

3 **Select the [Preview] tab to check the preview image.**
When a printer is connected, you can also check the image by printing.
For details, refer to "Print" on page 170.



NOTE	• Logo printing, barcode printing, 2D code printing, ESC command, buzzer sound, drawer kick, and paper cut are displayed as icons. • The layout may change depending on the preview settings. (For details, refer to "Setting" on page 167).

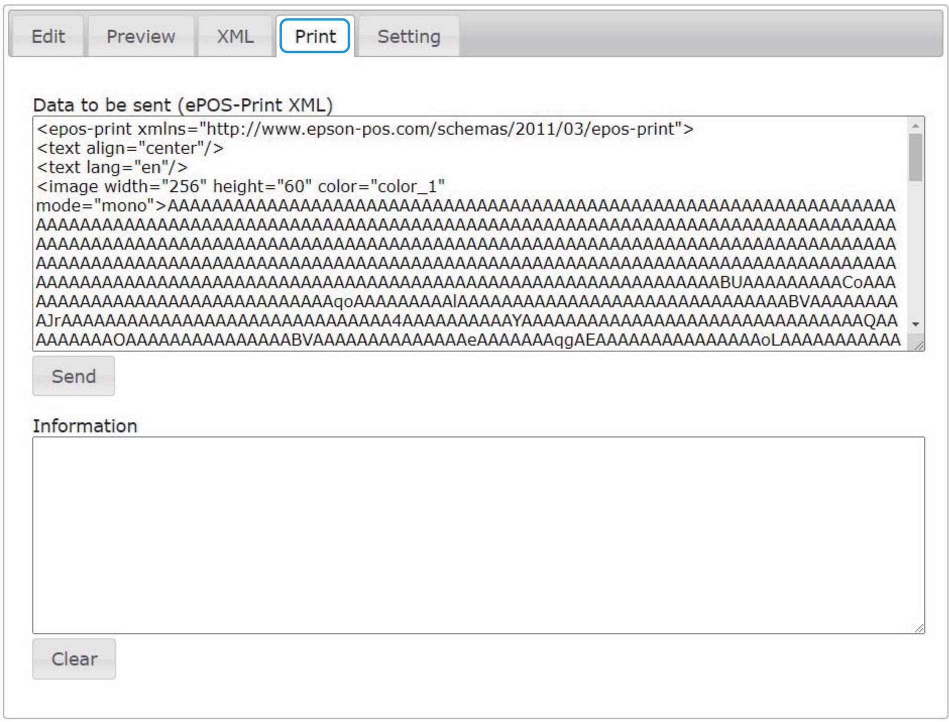
4 **Select the [XML] tab. The ePOS-Print XML document is displayed. Use it by copying.**
Save the ePOS-Print XML print document, and you can edit it by importing it again.



Print

Using the printer, print the ePOS-Print XML print document according to the printer's settings to perform test printing.

(For details on the printer settings, refer to "Setting" on page 167).



Item	Description
Data to be sent (ePOS-Print XML)	The ePOS-Print XML document is displayed.
Send	Sends data to the printer and performs printing.
Information	Displays the print status.
Clear	Deletes the content in the [Information] box.

Perform printing as follows:

1. **Select the [Print] tab.**

2. **Check the content in the [Data to be sent (ePOS-Print XML)] box and press the [Send] button.**

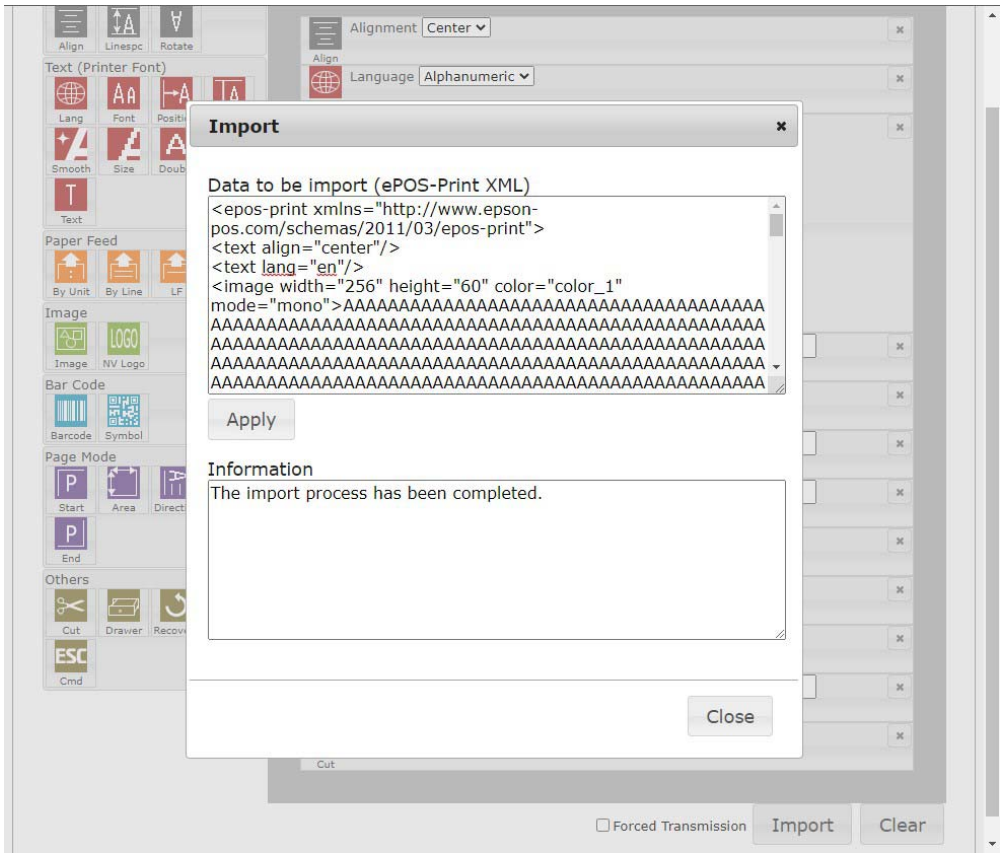
   The ePOS-Print XML print document created using the Edit tab page is displayed in "Data to be sent (ePOS-Print XML)".

3. **The print document is printed to TM-S9000II-NW.**
   The acquired status is displayed in the Information box.

Import

ePOS-Print Editor can import an ePOS-Print XML print document.

This function is used to edit the already created ePOS-Print XML print document.



Item	Description
Data to be import (ePOS-Print XML)	Pastes the ePOS-Print XML print document.
Apply	Imports the ePOS-Print XML print document.
Information	Displays the import information.
Close	Closes the Import screen.

ePOS-Print Editor can import an ePOS-Print XML print document as follows:

1 **Select the [Edit] tab and click the [Import] button.**

2 **The "Import" screen appears. Paste the ePOS-Print XML print document in the [Data to be import (ePOS-Print XML)] box.**

3 **Click the [Apply] button.**

4 **The "Confirmation" screen appears. Click the [Yes] button.**

Encoding Graphic Data

Use ePOS-Print Editor to convert raster graphic bit-image data into a base 64 encoded character string.

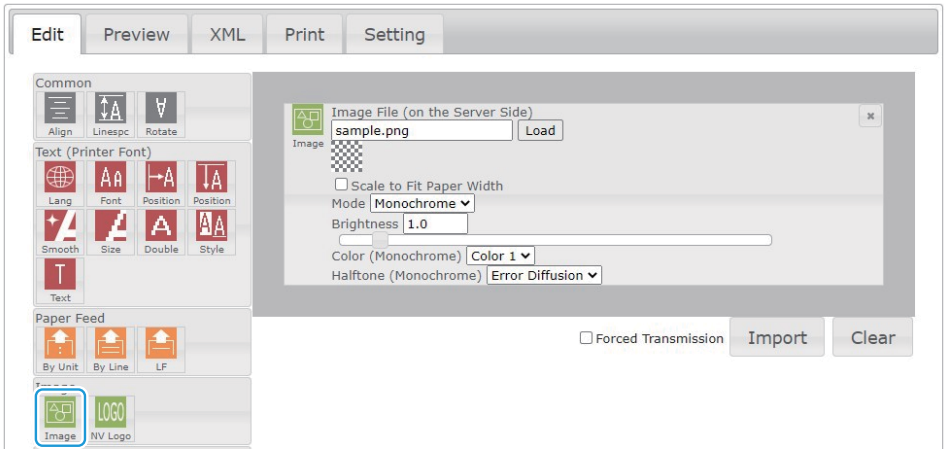
How to Use ePOS-Print Editor

Create data as follows:

1 **Store the image file you want to print into the same level as the index of this tool.**

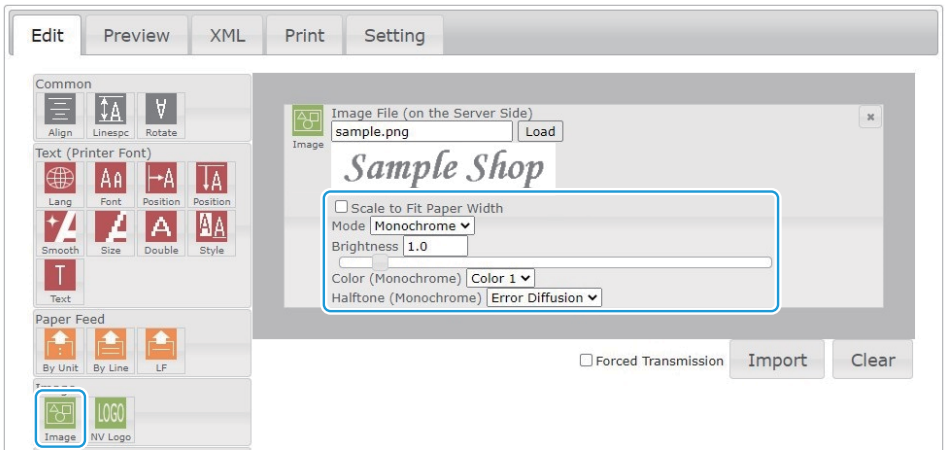
2 **Start ePOS-Print Editor.**

3 **Select the [Image] icon in the menu area and add it in the edit area.**



4 **Specify the name of the stored image file for "Image File (on the Server Side)" and click the [Load] button.**

5 **Specify values as needed for "Scale to Fit Paper Width", "Mode", "Brightness", "Color", and "Halftone".**



6 **Select the [XML] tab.**

7 **Code converted into a base 64 encoded character string is displayed in [ePOS-Print XML Print Document]. Copy the desired character string for use.**