

**ASK-1500/2000/2500
4000/4000A
Printer
Wrapper SDK Manual**

Ver.2.7.1.3

February 19, 2010

Copyright© 2005-2010 FUJIFILM Corporation

FUJIFILM

Revision history

Version	Date	Description
1.2.2.0	May 20, 2005	First release. This version is not supported the image processing function.
1.2.2.1	Jun. 7, 2005	Added Image Intelligence (I2) Ver.1.7. Sample program supports the image-quality setup process using I2 Ver.1.7. Printer 3D data file is changed.
1.3.0.0	Jun. 10, 2005	Added I2 functions. Changed the target printer to FUJIFILM ASK-1500. (INF file is changed only.) Error recovery process is changed. Error codes are changed. Added macro MCP_IS_ERRF_H_CONTINUE to judge recoverable error.
1.3.1.0	Jun. 24, 2005	Changed a default value of sharpness from 1 to 0. Changed 3d data file. (Bug Fix) When ink door open error has occurred on printing, SDK doesn't call callback function. (Bug Fix) When ink door open error has occurred on sending image, SDK doesn't notify ink door open error.
1.4.0.0	Jul. 4, 2005	Added functions for Color Correction Tool. Added error code MCP_ERR_NO_IMGSET_1DLUT_DATA.
1.5.0.0	Jul. 22, 2005	Added DPB-4000 as a target printer. Added multi print mode for two 4"x6" images using 6"x9" paper.
1.7.0.0	Sep. 21, 2005	(Bug Fix) When printing the image including white frame, the memory doesn't be release. (Bug Fix) When the SDK cannot allocate the memory, the error recovery has a problem. Usually, this problem doesn't occur.
1.9.0.0	Dec. 22, 2005	This version uses I2 Ver.2.5. Have two error recovery modes. Added new functions to have two error recovery modes. MCP_SetAutoRecoveryFlag(), MCP_StopPrint(). Added new function to set USB ID in Printer. MCP_SetUsbNo() Target printer name is changed from DPB-4000 to ASK-4000. Added ASKTool for changing USB ID.
2.1.0.0	Dec. 27, 2005	Supported ASK-4000 officially. Supported ASK-2000 with using only 4"x6". Changed the 3d file for ASK-4000. (Bug Fix)When the recoverable printing has occurred during printing some copies, SDK calls the callback function with different errors twice.
2.1.1.0	Jan. 16, 2006	Supported both ASCII code and UNICODE on DPBSMP.exe and FFISPL.DLL. Changed the 3d file for ASK-2000.
2.1.2.0	Feb. 6, 2006	Changed the dll MDPB.DLL to resolve communication errors. Changed the 3d file for ASK-2000. But this is not final release.
2.1.3.0	Feb. 10, 2006	Changed the dll MDPB.DLL to modify the color balance for only ASK-2000.
2.1.4.0	Feb. 21, 2006	Changed the 3d files for ASK-2000and ASK-4000.
2.3.0.0	Mar. 31, 2006	Changed the 3d file for ASK-4000. Changed the dll MDPB.DLL to modify the sharpness table. Changed the default value of sharpness for ASK-4000 (0->2) and ASK-2000 (0->1). Changed the paper remains value on MCP_GetPaperRemain for ASK-2000. Added MCP_GetPrinterError() to get the string of Printer LCD error and the detailed printer status. Added error code list in SDK. Changed the method to be recognized the device information by Windows.
2.3.0.1	Apr. 12, 2006	Changed the 3d file for ASK-2000.
2.3.0.2	May 16, 2006	Changed the dll MDPB.DLL to modify the color table for ASK-2000. Changed the 3d file for ASK-4000.
2.4.0.0	Aug. 14, 2006	Changed the paper remains value on MCP_GetPaperRemain for ASK-2000 (L, 2L, 6"x8", A5) and ASK-4000 (6"x12"). Change the dll MDPB.DLL to modify the processing of heat correction for print.

2.4.0.1	Aug. 22, 2006	Changed the dll (MDPB.DLL) to modify the paper remain value before first print.
2.4.1.0	Aug. 30, 2006	Added MCP_Generate1dDataWithGammaCurve(). This is the new algorithm to generate 1D LUT data.
2.4.2.0	Nov. 08, 2006	Changed the exclusive control for the sending of image. The sending of image is exclusively controlled between printers. (This version is only for Panda.)
2.4.3.0	Dec. 19, 2006	The handle leak of the thread for the print is corrected. Changed the dll (MDPB.DLL) to modify the processing of timeout for send image. (This version is only for Panda.)
2.4.7.0	May. 16, 2007	Changed the dll (MDPB.DLL) to strengthen the processing of communication. ➔ When the application shutdown illegally, the problem that cannot be communicated with printer was corrected. ➔ The problem (Windows USB driver's problem) that cannot be communicated with printer by disconnecting the USB communication was corrected. (This version is only for Panda.)
2.5.0.1	Jul. 20, 2007	Added division print mode for three 8"x4" images using 8"x12" paper. Added new function to print of division into three for ASK-4000. MCP_SendImageAndThreeDivPrint() (Refer to Section 7.3 for details.) Added ASK-4000A as a target printer (correspond to the print of the A4 size). Changed the dll (MDPB.DLL) to modify the processing of the gamma control. Additionally, changed the 3d file for ASK-4000 and ASK-4000A.
2.5.0.2	Sep. 6, 2007	Changed the Utility application (ASKTool.exe). ➔ It corresponded to ASK-4000A.
2.6.1.0	Oct. 5, 2007	Changed the dll (MDPB.DLL) to modify the processing of the sharpness. Changed the default value of sharpness for ASK-2000 in the dll (DPBAPI.DLL) and the initial file (CCORRECT.INI) (1 -> 4). Changed the 3D file for ASK-2000.
2.6.8.0	Oct. 9, 2008	Added ASK-2500 as a target printer. ➔ Printer ID returned by MCP_GetPrinterInfo() function is MCP_PRINTER_ID_ASK2500.
2.6.9.0	May. 1, 2009	Changed the dll (MDPB.DLL) to control ASK-2500. Changed the 3D file of normal mode for ASK-2500 and added the 3D file of high quality mode for ASK-2500. The rule of the file name changed when 1D file is created. (in MCP_Create1dDataFile) Before : PP1Dyyy.dat (yyy is printer serial number) After : PP1Dxxx_yyy.dat (xxx is printer name, yyy is printer serial number) For example, when printer is ASK-2000 and serial number is 090501333, 1D file name is "PP1D2000_090501333.dat".
2.6.9.1	Jul. 27, 2009	Changed the dll (MDPB.DLL) to control ASK-2500. ➔ The following sizes automatically change into the HQ mode at ASK-2500. 5"x7", 6"x8", 6"x9", 4"x6"x2(Multi Print mode) Changed the 3D file for ASK-2500.
2.6.9.2	Aug. 12, 2009	Changed the dll (MDPB.DLL) to control ASK-2500.
2.6.9.3	Aug. 20, 2009	Changed the dll (MDPB.DLL) to control ASK-2500.
2.6.9.4	Aug. 24, 2009	Changed the dll (MDPB.DLL) to control ASK-2500.
2.6.9.5	Sep. 16, 2009	Changed the dll (MDPB.DLL) to control ASK-2500. Changed the 3D file for ASK-2500.(DP3D2500.DAT and DP3D2500HQ.DAT)
2.6.9.6	Dec. 1, 2009	Changed the 3D file for ASK-2500.(DP3D2500.DAT and DP3D2500HQ.DAT)
2.7.1.0	Dec. 4, 2009	Changed the dll (DPBAPI.DLL). ➔ SDK discontinued the acquisition of printer information (serial number, printer id) in the HQ mode. Changed the dll (MDPB.DLL). ➔ The problem of mDPB_GetSerialNo() corrected.

2.7.1.1	Jan 13, 2010	Changed the 3D file for ASK-4000. (DP3D4000.dat) Changed the 3D file for ASK-4000A. (DP3D4000A.dat)
2.7.1.2	Jan 18, 2010	Changed the sample program (DPBSMP.exe) to correct the trouble of 3D file specification of HQ mode.
2.7.1.3	Feb 19, 2010	Changed the 3D file for ASK-2500.(DP3D2500.DAT and DP3D2500HQ.DAT) ➔ From 2009/11/27 version to 2009/11/23 version.

Table of Contents

1. Introduction	8
2. Overview of SDK	11
2.1 Printer Control State Transition	11
2.2 Coordinate System and Offset	11
2.3 Frame Memory of ASK-1500/2000/2500	11
2.3.1 In Case of 4"x6" Size	12
2.3.2 In Case of 6"x8" Size	13
2.3.3 In Case of 4"x6"x2 (Multi Print Mode)	14
2.4 Frame Memory of ASK-4000	14
2.4.1 In Case of 8"x4"x3 (Three Division Print Mode)	14
2.5 Frame Memory of ASK-4000A	14
2.6 Paper Size	15
3. Structures	16
3.1 MCP_PRINTER_INFO	16
3.2 MCP_PRINTINFO	17
3.3 MCP_PAPERINFO	17
3.4 MCP_IMAGEINFO	17
3.5 MCP_COLORINFO	18
3.6 MCP_PRINTINGINFO	18
3.7 MCP_VERSIONINFO	19
4. API Functions	20
4.1 MCP_Init	21
4.2 MCP_Finish	21
4.3 MCP_EnumPrinter	21
4.4 MCP_OpenPrinter	22
4.5 MCP_ClosePrinter	22
4.6 MCP_GetPrinterInfo	22
4.7 MCP_InitPrintSetting	23
4.8 MCP_SendPrintSetting	23
4.9 MCP_InitPrinterMemory	23
4.10 MCP_SendImageAndPrint	24
4.11 MCP_SendImageAndMultiPrint	25
4.12 MCP_SendImageAndThreeDivPrint	26
4.13 Callback Function	27
4.14 MCP_CancelPrint	27
4.15 MCP_ReceivePrintedInfo	27
4.16 MCP_ReceiveVerInfo	28

4.17 MCP_GetPaperRemain	28
4.18 MCP_GetLastError.....	28
4.19 MCP_LutConversionForPrinter	29
4.20 MCP_Generate1dData	29
4.21 MCP_Create1dDataFile	29
4.22 MCP_Delete1dDataFile	30
4.23 MCP_Exec1dConversion	30
4.24 MCP_IS_ERRF_H_CONTINUE.....	30
4.25 MCP_SetAutoRecoveryFlag	31
4.26 MCP_StopPrint.....	31
4.27 MCP_SetUsbNo.....	31
4.28 MCP_GetPrinterError.....	32
4.29 MCP_Generate1dDataWithGammaCurve.....	32
5. Image-quality setup process.....	34
5.1 Overview.....	34
6. Color Correction Tool	35
7. SDK Control Procedure.....	36
7.1 Flow of Print Process for ASK-1500/2000/2500/4000/4000A	37
7.1.1 Print Flow	37
7.1.2 Printing Information on Printing.....	39
7.2 Flow of Print Process for Multi Print Mode	40
7.2.1 Print Flow	40
7.2.2 Printing Information on Printing.....	41
7.3 Flow of Print Process for Three Division Print Mode.....	42
7.3.1 Print Flow	42
7.3.2 Printing Information on Printing.....	43
7.4 How to Control Two ASK-1500/2000/2500	44
8. SDK Error Code	45
8.1 API Function Error Code Details	45
8.1.1 Printer Condition Error (Flag:0x01 MCP_ERRF_P)	46
8.1.2 Printer Hardware Error (Flag:0x02 MCP_ERRF_H)	46
8.1.3 Communication Error (Flag:0x03 MCP_ERRF_I)	46
8.1.4 Software Error (Flag:0x04 MCP_ERRF_S).....	46
8.1.5 Printer Hardware Error (Flag:0x12 MCP_ERRF_H_CONTINUE)	47
8.2 MCP_GetLastError Code Details.....	48
8.2.1 Printer Condition Error (Flag:0x01 MCP_ERRF_P)	48
8.2.2 Printer Hardware Error (Flag:0x02 MCP_ERRF_H)	48
8.2.3 Printer Hardware Error (Flag: 0x12 MCP_ERRF_H_CONTINUE).....	49
8.2.4 Communication Error (Flag:0x03 MCP_ERRF_I)	49
8.2.5 Software Error (Flag:0x04 MCP_ERRF_S).....	49

8.3 Error Recovery	50
8.3.1 Printer.....	50
8.3.2 Application.....	51
8.4 How to Judge Recoverable or Non-recoverable Error	52
8.5 How to Get Error Code.....	52

- This SDK is provided based on the SDK License Agreement.
- The I2 module is also provided based on the I2 License Agreement.
- Windows is a registered trademark of Microsoft Corporation in the U.S. and elsewhere.
Windows is an abbreviation for Microsoft Windows Operating System.
- Other company names, product names, etc. are generally trademarks or registered trademarks of their respective companies.

1. Introduction

ASK-1500/2000/2500/4000/4000A Printer SDK (hereafter, the SDK) is a development kit that supports development of application software that controls FUJIFILM ASK-1500/2000/2500/4000/4000A Printer (hereafter, the Printer) from Windows personal computer to print images.

Using the SDK, you can create application software that makes the most of the Printer functions without troubling yourself with the details of a physical interface (such as USB) to which the Printer is connected.

The SDK provided is unique to a specific target printer.

● Target Printer

FUJIFILM ASK-1500 ... 3.5"x5", 4"x6", 5"x7", 6"x8", 6"x9", 4"x6"x2 (Multi Print Mode)

FUJIFILM ASK-2000 ...3.5"x5", 4"x6", 5"x7", 6"x8", 6"x9", 4"x6"x2 (Multi Print Mode)

FUJIFILM ASK-2500 ...3.5"x5", 4"x6", 5"x7", 6"x8", 6"x9", 4"x6"x2 (Multi Print Mode)

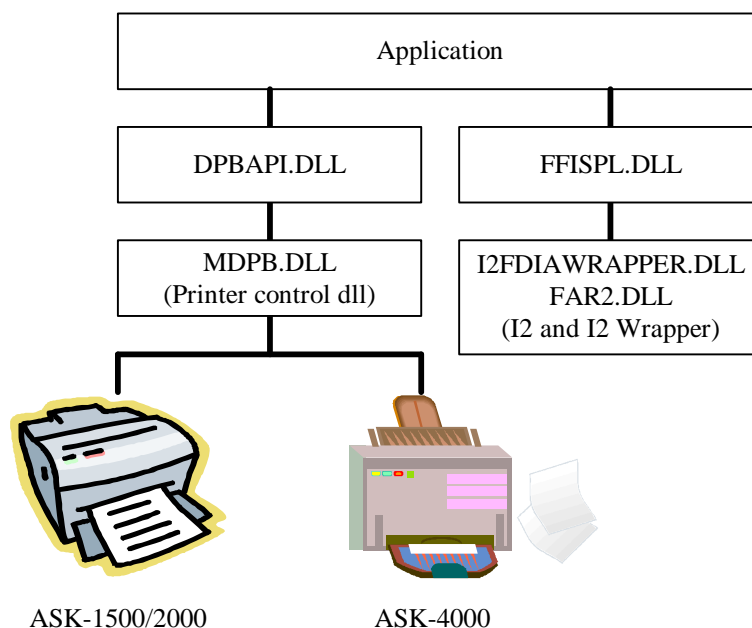
FUJIFILM ASK-4000 ... 8"x10", 8"x12", 8"x4"x3 (Three Division Print Mode)

FUJIFILM ASK-4000A ... A4

● File List

File Name	Version	Date	Description
DPBAPI.DLL	2.7.1.0	2009/12/04	Printer control DLL
DPBAPI.H	-	2008/10/09	Header file for printer control DLL
MDPB.DLL	3.2.1.0	2009/12/02	Lower Printer DLL for ASK-1500/2000/2500/4000/4000A
DPBSMP.EXE	2.7.1.0	2010/01/18	Sample program for ASK-1500/2000/2500/4000/4000A with ASCII code (with a source code)
CCORRECT.INI		2009/04/24	Settings file for DPBAPI.DLL
DP3D1500.DAT	-	2005/06/21	3D data file for ASK-1500
DP3D2000.DAT	-	2007/08/27	3D data file for ASK-2000
DP3D2500.DAT	-	2009/11/23	3D data file for ASK-2500
DP3D4000.DAT	-	2010/01/08	3D data file for ASK-4000
DP3D4000A.DAT	-	2010/01/08	3D data file for ASK-4000A
DP3D2000HQ.DAT		2007/10/18	3D data file for ASK-2000 (For High Quality mode)
DP3D2500HQ.DAT		2009/11/23	3D data file for ASK-2500 (For High Quality mode)
FFISPL.DLL	1.4.2.0	2006/04/22	I2 Wrapper with ASCII code
FFISPL.H	-	2006/03/31	Header file for I2 Wrapper
I2FDIAWRAPPER.DLL	2.5.0.3	2005/10/28	I2 related files

FAR2.DLL	2.0.0.0	2006/04/19	
LIBMMD.DLL	-	2003/10/19	
Param	-	-	Parameters folder for I2
MANUAL.PDF	-	2010/02/19	Manual
ISPL_SPEC_V142.PDF	-	2006/04/24	I2 wrapper Ver.2.5 specification
ERRORCODELIST.XLS	-	2006/03/13	Error code list
DEV_RECOGNITION.PDF	-	2006/03/30	Explanation of Device Recognition
ASKTOOL.EXE	1.6.0.0	2008/10/09	Utility software to change USB ID.



- Operating System
Windows® 2000 Professional
Windows® XP Home Edition/Professional

- Recommended development environment
Microsoft Visual C++® 6.0 (SP4 or later)

- Environment in which the operation is guaranteed

Items	Description
CPU ^{*1}	Intel Pentium 4 2.0GHz or more
Memory ^{*2}	512MB or more
USB speed	USB 2.0 only
Maximum number of printers that can be used simultaneously	3 printers ^{*3}

^{*1} Some functions in the SDK need much CPU power. Therefore, each processing time changes according to the capability of CPU.
If possible, it is better to choose faster CPU.

*² Memory size depends on the functions that application has.

*³ **USB ID should be set the unique number for connecting all printers. When changing USB ID, use ASKTool.**

If you use the SDK in an environment other than the one in which the operation is guaranteed, you will be bound by the stipulations in the **SDK License Agreement** separately concluded. When using I2 module, you should conclude **I2 License Agreement**.

- How to recognize the device information on Windows

There are two ways to be recognized the device information by Windows.

- ✧ Use INF file
- ✧ Use Printer Driver

When you will establish a system with some printers, you should use Printer Driver to be recognized the device information.

Because when using INF file, Windows will sometimes show the new device wizard after turning on a Printer.

You should use Printer Driver for only recognition on Windows.

Please refer “DEV_RECOGNITION.PDF” attached in SDK.

2. Overview of SDK

2.1 Printer Control State Transition

To perform printer control using the SDK, you need to control the following state transition. Each of the API functions provided by the SDK can be used only in a fixed state.

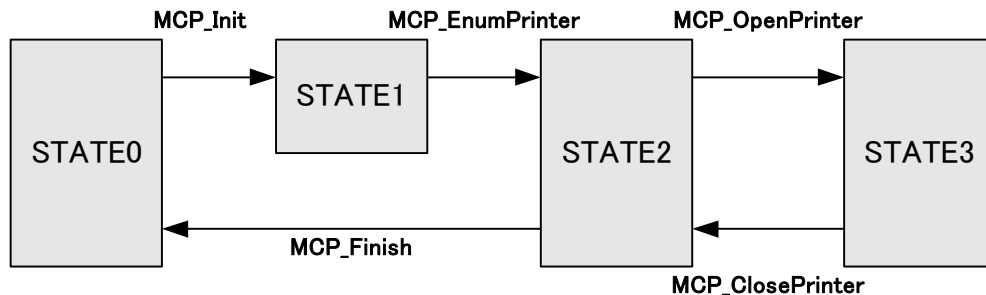


Figure 1: State Transition in the SDK

State name	Details
STATE0	Only the SDK dll is loaded. No printer control can be performed.
STATE1	Printer enumeration is incomplete. The SDK cannot be used.
STATE2	The SDK has been initialized.
STATE3	A communication connection with the Printer is established. Printer control can be performed.

2.2 Coordinate System and Offset

To print image data, you need to specify where it should be placed on paper (Printing area). This is called an offset value. The following explains how to calculate an offset value for each of the printer models.

2.3 Frame Memory of ASK-1500/2000/2500

Frame memory of ASK-1500/2000/2500 is as follows.

At this time, the main scanning direction (the direction of a printer head) is defined as the X-axis, and the sub scanning direction (the direction of paper delivery) is defined as the Y-axis.

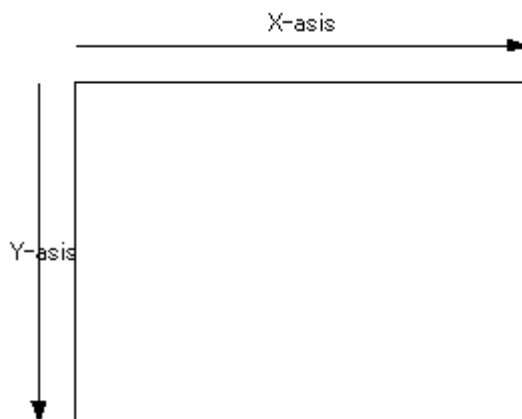


Figure 2: Frame Memory of ASK-1500/2000/2500 Printer

ASK-1500/2000/2500 can be printed in the paper of 3.5"x5"(L), 4"x6"(K), 5"x7"(2L), 6"x8" and 6"x9" size.

Therefore, image data size for the main scanning direction and the sub scanning direction are as follows.

	3.5"x5"	4"x6"	5"x7"	6"x8"	6"x9"	4"x6"x2
Main scanning direction	1536	1832	1536	1832	1832	1832
Sub scanning direction	1074	1228	2130	2432	2748	1228

In the case of printing in the paper of 4"x6" size, the image data size needs to be 1832 x 1228 pixels.

If it is not fulfill this condition, executes the following processes.

*Image data is rotated 90 degrees.

*Image data is expand or reduced to fit the size.

2.3.1 In Case of 4"x6" Size

For example, consider the case that image data (Width: 1832pixels, Height: 1228pixels) as shown in Figure 3 is printed.



Figure 3:Image data (Width: 1832pixels, Height: 1228pixels)

At this time, the printing conditions (MCP_PAPERINFO structure and MCP_IMAGEINFO structure) are as follows.

Structure	Member	Value	Notes
Paper info. (MCP_PAPERINFO)	Paper size	PaperSizeX=1832	
		PaperSize.Y=1228	
Image data info. (MCP_IMAGEINFO)	Image data size	ImageData.X=1832	
		ImageData.Y=1228	
	Offset value	Offset.X=0	Offset value must be set 0.
		Offset.Y=0	

2.3.2 In Case of 6"x8" Size

For example, consider the case that image data (Width: 2400pixels, Height: 1800pixels) as shown in Figure 4 is printed.



Figure 4: Image data (Width: 2400pixels, Height: 1800pixels)

In this case, because the width of image data is not settled in the main scanning direction, it is necessary to rotate image data 90 degrees as shown in Figure 5.



Figure 5: Image data after rotation and expansion (Width: 1832pixels, Height: 2432pixels)

At this time, the printing conditions (MCP_PAPERINFO and MCP_IMAGEINFO) are as follows.

Structure	Member	Value	Notes
Paper info. (MCP_PAPERINFO)	Paper size	PaperSizeX=1832	
		PaperSize.Y=2432	
Image data info. (MCP_IMAGEINFO)	Image data size	ImageData.X=1832	
		ImageData.Y=2432	
	Offset value	Offset.X=0	Offset value must be set 0.
		Offset.Y=0	

2.3.3 In Case of 4"x6"x2 (Multi Print Mode)

ASK-1500/2000/2500 can be printed two 4"x6" images on the paper of 6"x9" size at the same time.

At this time, the printing conditions (MCP_PAPERINFO and MCP_IMAGEINFO) are as follows.

In this case, PaperSize.Y is same as ImageData.Y.

Structure	Member	Value	Notes
Paper info. (MCP_PAPERINFO)	Paper size	PaperSizeX=1832	
		PaperSize.Y=1228	
Image data info. (MCP_IMAGEINFO)	Image data size	ImageData.X=1832	
		ImageData.Y=1228	
	Offset value	Offset.X=0	Offset value must be set 0.
		Offset.Y=0	

2.4 Frame Memory of ASK-4000

ASK-4000 can be printed in the paper of 8"x10" and 8"x12" size.

Therefore, image data size for the main scanning direction and the sub scanning direction are as follows.

	8"x10"	8"x12"	8"x4"x3
Main scanning direction	2444	2444	2444
Sub scanning direction	3044	3644	1244

2.4.1 In Case of 8"x4"x3 (Three Division Print Mode)

ASK-4000 can be printed three 8"x4" images on the paper of 8"x12" size at the same time.

At this time, the printing conditions (MCP_PAPERINFO and MCP_IMAGEINFO) are as follows.

In this case, PaperSize.Y is same as ImageData.Y.

Structure	Member	Value	Notes
Paper info. (MCP_PAPERINFO)	Paper size	PaperSizeX=2444	
		PaperSize.Y=1244	
Image data info. (MCP_IMAGEINFO)	Image data size	ImageData.X=2444	
		ImageData.Y=1244	
	Offset value	Offset.X=0	Offset value must be set 0.
		Offset.Y=0	

2.5 Frame Memory of ASK-4000A

ASK-4000A can be printed in the paper of A4 size.

Therefore, image data size for the main scanning direction and the sub scanning direction are as follows.

	A4
Main scanning direction	2528
Sub scanning direction	3556

2.6 Paper Size

The paper size that can be printed on ASK-1500/2000/2500/4000/4000A is as follows.

And the value of MCP_PAPERINFO structure and MCP_IMAGEINFO structure object at that time are as follows.

It cannot be printed in other paper sizes.

Moreover, the image data that does not match this size cannot be printed either.

When the image data to print is sizes other than the above, you have to fill a short part by the white frame, or to expand or reduce image data.

	ASK-1500/2000/2500				ASK-4000		ASK-4000A
Structure member	3.5"x5"	4"x6"	5"x7"	6"x8"	8"x10"	8"x12"	A4
PaperSize.X ImageData.X	1536	1832	1536	1832	2444	2444	2528
PaperSize.Y ImageData.Y	1074	1228	2130	2432	3044	3644	3556

The following sizes automatically change into the HQ mode at ASK-2500.

5"x7", 6"x8", 6"x9", 4"x6"x2(Multi Print mode)

3. Structures

Structure	Details
MCP_PRINTER_INFO	Printer device information
MCP_PRINTINFO	Print information
MCP_PAPERINFO	Paper Information
MCP_IMAGEINFO	Image data information
MCP_COLORINFO	Color correction information
MCP_PRINTINGINFO	Printing information
MCP_VERSIONINFO	Software version information

3.1 MCP_PRINTER_INFO

```
[Definition] typedef struct {
    short    nUsbNo;
    short    reserved;
    short    nPrinterId;
    short    nMediaType;
    char     szSerialNumber[64];
} MCP_PRINTER_INFO
```

[Description] Printer information

Member	Type	Explanation
nUsbId	short	Number of USB connection
nPrinterId	short	ID number for each printer MCP_PRINTER_ID_ASK1500 MCP_PRINTER_ID_ASK2000 MCP_PRINTER_ID_ASK2500 MCP_PRINTER_ID_ASK4000 MCP_PRINTER_ID_ASK4000A MCP_PRINTER_ID_UNKNOWN
nMediaType	short	Paper media type MCP_MEDIA_TYPE_L MCP_MEDIA_TYPE_K MCP_MEDIA_TYPE_2L MCP_MEDIA_TYPE_6X8_A5 MCP_MEDIA_TYPE_K_MULTI MCP_MEDIA_TYPE_8X10 MCP_MEDIA_TYPE_8X12 MCP_MEDIA_TYPE_UNKNOWN
szSerialNumber	char	Printer serial number

3.2 MCP_PRINTINFO

```
[Definition]  typedef struct {
                WORD  wCopies;
                MCP_PAPERINFO    PaperInfo;
                MCP_IMAGEINFO    ImageInfo
                MCP_COLORINFO    ColorInfo;
                short  nPrintMode;
            } MCP_PRINTINFO
```

[Description] Print information

Member	Type	Explanation
wCopies	WORD	Print copy count
PaperInfo	MCP_PAPERINFO	Paper information
ImageInfo	MCP_IMAGEINFO	Image data information
ColorInfo	MCP_COLORINFO	Color correction information
nPrintMode	short	Print mode MCP_PRINT_MODE_SPEED MCP_PRINT_MODE_QUALITY (for ASK-2000 and ASK-2500)

3.3 MCP_PAPERINFO

```
[Definition]  typedef struct {
                POINT    PaperSize;
            } MCP_PAPERINFO
```

[Description] Paper information

Member	Type	Explanation
PaperSize	POINT	Print paper size

3.4 MCP_IMAGEINFO

```
[Definition]  typedef struct {
                POINT    ImageData;
                POINT    Offset;
            } MCP_IMAGEINFO
```

[Description] Image data information

Member	Type	Explanation
ImageData	POINT	Image size
Offset	POINT	Output offset values X and Y 0 fixed

3.5 MCP_COLORINFO

[Definition] typedef struct {
 short nSharpness;
 int nReserved;
 } **MCP_COLORINFO**

[Description] Color correction information

Member	Type	Explanation
nSharpness	short	Sharpness parameter for printer. MCP_SHARPNESS_DEFAULT MCP_SHARPNESS_NOCORRECT MCP_SHARPNESS_DEFAULT_1500 MCP_SHARPNESS_DEFAULT_4000 MCP_SHARPNESS_DEFAULT_2000 MCP_SHARPNESS_DEFAULT_4000A MCP_SHARPNESS_DEFAULT_2500

[Notes] The current default value for each printer is as follows.

ASK-1500 : 0
 ASK-4000 : 2
 ASK-2000 : 4
 ASK-4000A : 2
 ASK-2500 : 4 (TBD)

3.6 MCP_PRINTINGINFO

[Definition] typedef struct {
 BYTE bDetailStatus;
 WORD wJobNumber;
 WORD wPrintedNumber;
 WORD wPrintRequestNumber;
 } **MCP_PRINTINGINFO**

[Description] Printing information

Member	Type	Explanation
bDetailStatus	BYTE	Detail printer status MCP_PRINTERSTATUS_READY MCP_PRINTERSTATUS_PRINTING MCP_PRINTERSTATUS_INITIALIZE MCP_PRINTERSTATUS_JOB_CANCEL MCP_PRINTERSTATUS_START_SENDING MCP_PRINTERSTATUS_SEND_COMPLETE
wJobNumber	WORD	Print job number
wPrintedNumber	WORD	Printed copies
wPrintRequestNumber	WORD	Print request copies

3.7 MCP_VERSIONINFO

[Definition] typedef struct {
 char szVersion1[64];
 char szVersion2[64];
 char szVersion3[64];
 } **MCP_VERSIONINFO**

[Description] Software version information

[Members]	Member	Type	Explanation
	szVersion1	char	Firmware version
	szVersion2	char	Not used
	szVersion3	char	Not used

4. API Functions

API function	Details
MCP_Init	Initialize the SDK.
MCP_Finish	Exit the SDK.
MCP_EnumPrinter	Search target printers.
MCP_OpenPrinter	Start to control a target printer..
MCP_ClosePrinter	Close to control a target printer.
MCP_GetPrinterInfo	Acquire printer device information
MCP_InitPrintSetting	Initialize SDK parameters for print setting area with default values.
MCP_SendPrintSetting	Send print setting data to a target printer.
MCP_InitPrinterMemory	Clear frame memory on printer.
MCP_SendImageAndPrint	Send image data to a target printer, and start to print.
MCP_ReceivePrintedInfo	Acquire printer cumulative printed count.
MCP_ReceiveVerInfo	Acquire version information for printer.
MCP_GetPrinterStatus	Acquire current status of a target printer.
MCP_GetPaperRemain	Acquire information for paper remains.
MCP_GetLastError	Acquire the detailed error code.
MCP_CancelPrint	Stop printing.
MCP_LutConversionForPrinter	Perform 3D LUT conversion.
MCP_SetPrinterMemoryClearFlag	Set the mode to clear frame memory on printer.
MCP_Generate1dData	Generate 1d data to use DCMY parameters that Operator inputs such like a mini-lab system.
MCP_Generate1dDataWithGammaCurve	Generate 1d data to use DCMY parameters. This is a data to increase or decrease value of middle range.
MCP_Create1dDataFile	Create 1d data file that corresponds to a printer. This is similar with FRONTIER key adjustment.
MCP_Delete1dDataFile	Delete 1d data file that corresponds to a printer.
MCP_Exec1dConversion	Perform 1d LUT conversion to the specified image data.
MCP_IS_ERRF_H_CONTINUE	Judge recoverable or non-recoverable error.
MCP_SetAutoRecoveryFlag	Set the flag to indicate the way of the error recovery.
MCP_StopPrint	Delete image in Printer when error has occurred.
MCP_SetUsbNo	Set the USB NO(ID) to have the unique number.
MCP_GetPrinterError	Get the string of Printer LCD error and the detailed printer status.

4.1 MCP_Init

[Function]	DWORD MCP_Init(void)
[Description]	Initialize the SDK.
[Arguments]	None
[Return Value]	MCP_OK : Success Other: Error
[Details]	Declares the start of using the SDK. Upon declaration, this function initializes the SDK and moves to STATE1 .
[State]	This function is enabled in STATE0 .

4.2 MCP_Finish

[Function]	DWORD MCP_Finish(void)
[Description]	Exit the SDK.
[Arguments]	None
[Return Value]	MCP_OK : Success Other: Error
[Details]	Declares the end of using the SDK. Upon declaration, the function exits the SDK and moves to STATE0 .
[State]	This function is enabled in STATE2 .

4.3 MCP_EnumPrinter

[Function]	DWORD MCP_EnumPrinter(PUINT puiNumPrinter, PHANDLE phPrinter)		
[Description]	Find and list printers to be controlled.		
[Arguments]	puiNumPrinter	IN	Maximum number of printer handles that can be acquired.
		OUT	Number of printer handles that can be found.
	phPrinter	OUT	Pointer of an array in which printer handles should be stored.
[Return Value]	MCP_OK : Success Other: Error		
[Details]	Find and list ASK-1500/2000/2500/4000/4000A printers. When your system has some ASK-1500/2000/2500/4000/4000A printers, your application has to find them at one time. In the SDK, application cannot find them separately. This function moves to STATE2 .		
[State]	This function is enabled in STATE1 .		

4.4 MCP_OpenPrinter

[Function] DWORD MCP_OpenPrinter(HANDLE hPrinter, DWORD dwIdleTimeout)

[Description] Open a connection to a target printer.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	dwIdleTimeout	IN	MCP_TIMEOUT_INFINITE fixed

[Return Value] **MCP_OK**: Success

Other: Error

[Details] The SDK doesn't have the timeout function.

When this function normally ends, the SDK is moved to **STATE3**.

[State] This function is enabled in **STATE2**.

4.5 MCP_ClosePrinter

[Function] DWORD MCP_ClosePrinter(HANDLE hPrinter)

[Description] Close a connection with a target printer.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
-------------	----------	----	--------------------------------------

[Return Value] **MCP_OK**: Success

Other: Error

[Details] When this function normally ends, the SDK is moved to **STATE2**.

[State] This function is enabled in **STATE3**.

4.6 MCP_GetPrinterInfo

[Function] DWORD MCP_GetPrinterInfo(HANDLE hPrinter,
PMCP_PRINTER_INFO pPrinterInfo)

[Description] Acquire the detailed information of a target printer in the SDK.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	pPrinterInfo	OUT	Detailed information of a target printer.

[Return Value] **MCP_OK**: Success

Other: Error

[Details] Acquire the detailed information of a target printer obtained in an inquiry data in the SDK.

All data stored in pPrinterInfo is valid.

[State] This function is enabled in **STATE3**.

4.7 MCP_InitPrintSetting

[Function] DWORD MCP_InitPrintSetting(HANDLE hPrinter)

[Description] Initialize the print setting information held in the SDK.

[Arguments]

hPrinter	IN	Printer handle for a target printer.
----------	----	--------------------------------------

[Return Value] **MCP_OK**: Success

Other: Error

[Details] Application calls this function after **MCP_OpenPrinter** and doesn't call it at every print sequence.

The following values are set in the function.

Copies	1
Paper size	1536x1074 (L)
Image data size	
Offset	(0, 0)
Sharpness	ON (1)

[State] This function is enabled in **STATE3**.

4.8 MCP_SendPrintSetting

[Function] DWORD MCP_SendPrintSetting(HANDLE hPrinter,
MCP_PRINTINFO PrintInfo)

[Description] Send the print setting information to a target printer.

[Arguments]

hPrinter	IN	Printer handle for a target printer.
PrinterInfo	IN	Print setting information.

[Return Value] **MCP_OK**: Success

Other: Error

[Details] **In MCP_PRINTINFO structure, the SDK can be set the sharpness parameter.****It is recommended to set the specified value as the sharpness parameter.****ASK-1500 : 0, ASK-4000 : 2, ASK-2000 : 4,****ASK-4000A : 2, ASK-2500 : 4 (TBD)****The SDK doesn't use the parameters of color correction except sharpness.****The image-quality setup process has some parameters of color correction.**[State] This function is enabled in **STATE3**.

4.9 MCP_InitPrinterMemory

[Function] DWORD MCP_InitPrinterMemory(HANDLE hPrinter)

[Description] Clear the frame memory on a target printer.

[Arguments]

hPrinter	IN	Printer handle for a target printer.
----------	----	--------------------------------------

[Return Value] **MCP_OK**: Success

Other: Error

[Details] Usually application doesn't call this function.

[State] This function is enabled in **STATE3**.

4.10 MCP_SendImageAndPrint

[Function] DWORD MCP_SendImageAndPrint(HANDLE hPrinter,
HANDLE hDib, PMCP_CALLBACK pCallbackProc)

[Description] Send image data to a target printer, instruct to starting print and monitor the progress of printing.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	hDib	IN	Image data handle to send. This is memory handle later than BITMAPINFOHEADER of DIB.
	pCallbackProc	IN	Address of the progress notification callback function. Set NULL, if no notification is required.

[Return Value] **MCP_OK**: Success

Other: Error

[Details] Send image data to a target printer.
After sending image data, printing is automatically started.
Then, the callback function is called to report the status to an application.
When application receives **MCP_PRINTERSTATUS_START_SENDING** via callback function, it can start to perform the process for the next image.

While printing is in progress, the SDK checks the printer status periodically (every 0.2 seconds) to monitor the end of printing.

When printing of one sheet is completed, the callback function is called to report the status and print processing progress to an application.

The above procedure is repeated until all copies have been printed.

The data format supported by this function is only DIB of 24 bits.

This function immediately returns control after image data transfer is started.

Because it operates by creating a new thread.

[State] This function is enabled in **STATE3**.

4.13 Callback Function

[Function] void CallbackProc(HANDLE hPrinter,

PMCP_PRINTINGINFO pPrintingInfo, DWORD dwErrorInfo)

[Description] When printer's status is changed, the SDK calls this callback function.

[Arguments]	hPrinter	OUT	Printer handle for a target printer.
	pPrintingInfo	OUT	Printing progress information.
	dwErrorInfo	OUT	Error information.

[Details] This assignment of bDetailStatus of MCP_PRINTINGINFO structure is as follows.

MCP_PRINTERSTATUS_READY**MCP_PRINTERSTATUS_PRINTING****MCP_PRINTERSTATUS_INITIALIZE****MCP_PRINTERSTATUS_JOB_CANCEL****MCP_PRINTERSTATUS_START_SENDING****MCP_PRINTERSTATUS_SEND_COMPLETE****MCP_PRINTERSTATUS_START_RECOVERY**

4.14 MCP_CancelPrint

[Function] DWORD MCP_CancelPrint(HANDLE hPrinter)

[Description] Cancel printing on a target printer.

[Arguments]	hPrinter	OUT	Printer handle for a target printer.
-------------	----------	-----	--------------------------------------

[Return Value] **MCP_OK**: Success

Other: Error

[Details] The print operation is stopped after the print processing currently in progress (printing in yellow) ends.

[State] This function is enabled in **STATE3**.

4.15 MCP_ReceivePrintedInfo

[Function] DWORD MCP_ReceivePrintedInfo(HANDLE hPrinter,
PDWORD pPrintedInfo)

[Description] Acquire the cumulative printed count held on a target printer.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	pdwTotalPrintCount	OUT	Printer cumulative printed count after shipment.

[Return Value] **MCP_OK**: Success

Other: Error

[State] This function is enabled in **STATE3**.

4.16 MCP_ReceiveVerInfo

[Function] DWORD MCP_ReceiveVerInfo(HANDLE hPrinter,

PMCP_VERSIONINFO pVersionInfo)

[Description] Acquire the software version and parameter version of a target printer.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	pVersionInfo	OUT	Software version and parameter version.

[Return Value] **MCP_OK**: Success

Other: Error

[State] This function is enabled in **STATE3**.

4.17 MCP_GetPaperRemain

[Function] DWORD MCP_GetPaperRemain(HANDLE hPrinter,

PDWORD pdwPaperRemain)

[Description] Acquire the information of paper remain.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	pdwPaperRemain	OUT	Information of paper remain

[Return Value] **MCP_OK**: Success

Other: Error

[Details] This value isn't exact and a rough value.

[State] This function is enabled in **STATE3**.

4.18 MCP_GetLastError

[Function] DWORD MCP_GetLastError(HANDLE hPrinter)

[Description] Acquire the detailed error code of an error that occurred in the SDK.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
-------------	----------	----	--------------------------------------

[Return Value] Error code

[State] This function is enabled in all states.

4.19 MCP_LutConversionForPrinter

[Function] DWORD MCP_LutConversionForPrinter(HANDLE hDib,
PCSTR strDataFileName)

[Description] Convert color space from sRGB to printer.

[Arguments]	hDib	IN	Image handle. This is the memory handle later than BITMAPINFOHEADER of DIB.
	strDataFileName	IN	Data file name for printer 3d LUT. This is only file name, not included file path.

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] **Application has to call this function after image-quality setup process.**

[State] This function is enabled in **STATE3**.

4.20 MCP_Generate1dData

```
[Function]    DWORD MCP_GenerateIdData(LPBYTE lpbData,
                                         int nDdata, int nCdata, int nMdata, int nYdata)
```

[Description]	Generate 1d data to use DCMY parameters that Operator inputs such like a mini-lab system.
---------------	---

[Arguments]	lpbData	OUT	Memory to hold 1d data.
	nDdata	IN	D parameter.
	nCdata	IN	C parameter.
	nMdata	IN	M parameter
	nYdata	IN	Y parameter.

[Return Value] **MCP OK:** Success

Other: Error

[Explanation] Allocate memory to hold 1d data by application.

[State] This function is enabled in **STATE3**.

4.21 MCP Create1dDataFile

[illegible]

[Description] Create 1d data file that corresponds to a printer.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	lpbData	IN	Memory to hold 1d data.
	szFileName	IN	Directory to save ad data file.
		OUT	1d data file name. (Full Path)

[Return Value]	MCP_OK: Success
----------------	------------------------

Other: Error

- [Explanation] Put 1d data into lpbData to call MCP_Generate1dData() before this function.
- [State] This function is enabled in **STATE3**.
- [Other] Created file name is PP1Dxxx_yyy.dat. xxx is printer name. yyy is printer serial number. (For example, when printer is ASK-2000 and serial number is 090501333, 1D file name is "PP1D2000_090501333.dat".)

4.22 MCP_Delete1dDataFile

[Function] DWORD MCP_Delete1dDataFile(HANDLE hPrinter, char *szFileName)

[Description] Delete 1d data file that corresponds to a printer.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	lpbData	IN	Memory to hold 1d data.
	szFileName	IN	Directory where 1d data file saved
		OUT	Deleted 1d data file name. (Full Path)

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] When SDK can't delete 1d data file, this function returns NULL as szFileName.

[State] This function is enabled in **STATE3**.

4.23 MCP_Exec1dConversion

[Function] DWORD MCP_Exec1dConversion(HANDLE hPrinter, HANDLE hDib)

[Description] Perform 1d LUT conversion to the specified image data.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	hDib	IN	DIB handle for image data
		OUT	

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] Application calls this function to perform 1d LUT conversion.
Set the memory handle after BITMAPINFOHEADER of DIB to hDib.
This function can accept only DIB of 24bit.
In this function, it is called MCP_Search1dDataFile() and MCP_Do1dLut().

[State] This function is enabled in **STATE3**.

4.24 MCP_IS_ERRF_H_CONTINUE

[Function] BOOL MCP_IS?ERRF_H_CONTINUE(DWORD dwErrorCode)

[Description] Judge recoverable or non-recoverable error using return value of MCP_GetLastError().

[Arguments]	dwErrorCode	IN	Error code of MCP_GetLastError().
-------------	-------------	----	-----------------------------------

[Return Value] TRUE: Recoverable error

FALSE: Non-recoverable error

4.25 MCP SetAutoRecoveryFlag

[illegible]

[Description] Set the flag to indicate the way of the error recovery.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	boRecoveryFlag	IN	TRUE: Error recovery in printer. FALSE: Error recovery in application

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] This flag is to indicate the way of the recovery procedure when recoverable error has occurred.

When recoverable error has occurred with TRUE setting, printer automatically prints images in printer memory after changing paper or ink.

When recoverable error has occurred with FALSE setting, application should re-send images in printer memory after changing paper or ink. Because the SDK automatically removes images in printer memory after error has occurred. The SDK has the default value FALSE as this flag.

This flag is available from MCP_OpenPrinter() to MCP_ClosePrinter().

4.26 MCP_StopPrint

[Function] DWORD MCP_StopPrint(HANDLE hPrinter)

[Description] Delete images in printer when the error has occurred.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
-------------	----------	----	--------------------------------------

[Return Value] **MCP OK:** Success

Other: Error

[Explanation] Application should use the function, when application wants to stop the recovery procedure of printer.

This function removes images in printer memory.

4.27 MCP_SetUsbNo

[Function] DWORD MCP_SetUsbNo(HANDLE hPrinter, int nUsbNo)

[Description] Set the USB NO(ID) to have the unique number.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	nUsbNo	IN	USB NO(ID)

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] When using several printers in application at the same time, please set the unique USB NO for each printer.

4.28 MCP_GetPrinterError

[Function] **DWORD MCP_GetPrinterError(HANDLE hPrinter, char *szPrinterError, DWORD *dwPrinterErrorDetail)**

[Description] Get the string of Printer LCD error and the detailed printer status.

[Arguments]	hPrinter	IN	Printer handle for a target printer.
	szPrinterError	OUT	String of Printer LCD error
	dwPrinterErrorDetail	OUT	Detailed printer status

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] When error has occurred, application should call this function after MCP_GetLastError().

[Notes] * Application should allocate 4 bytes memory for szPrinterError before calling this function.

* If error is not printer error, the string of Printer LCD error is NULL.

* If printer is ASK-1500, the string of Printer LCD error is NULL.

Please refer "ERRORCODELIST.XLS" attached in SDK.

[Detailed printer status]

Detailed printer status is allocated the following information for each bit.

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	Recovery Processing	USB error	Communication error	Command error	Parameter error
b15	b14	b13	b12	b11	b10	b9	b8
Buffer full	Buffer1 in use	Buffer empty	Head cooling	Head warming	System error	No paper	No ink
b7	b6	b5	b4	b3	b2	b1	b0
Motor/Sensor error	Paper door open	Ribbon mark error	Ribbon door open	Paper end	Ribbon end	Paper jamming	Mismatch combination

4.29 MCP_Generate1dDataWithGammaCurve

[Function] **DWORD MCP_Generate1dDataWithGammaCurve(LPBYTE lpbData, int nDdata, int nCdata, int nMdata, int nYdata)**

[Description] Generate 1d data to use CMY parameters. This is a data to increase or decrease value of middle range. 0 and 255 is fixed. D is not used.

[Arguments]	lpbData	OUT	Memory to hold 1d data.
-------------	---------	-----	-------------------------

nDdata	IN	D parameter. (0 fixed)
nCdata	IN	C parameter.
nMdata	IN	M parameter
nYdata	IN	Y parameter.

[Return Value] **MCP_OK**: Success

Other: Error

[Explanation] Allocate memory to hold 1d data by application.

[State] This function is enabled in **STATE3**.

5. Image-quality setup process

We are preparing the Image Intelligence Ver.2.5.

Please refer FFISPL library specification.

5.1 Overview

Image-quality setup process can be executed both automatically and manually. Outline of each process is as follows.

<Auto image-quality setup process>

Auto image-quality setup process improves the quality of an image data which is pictured by the digital camera.

For an amusement use, the mode that reduces effects to suitable level is also provided.

<Manual image-quality setup process>

Manual image-quality setup process provides the method of manual color correction that can be set up freely for image data that is already processed.

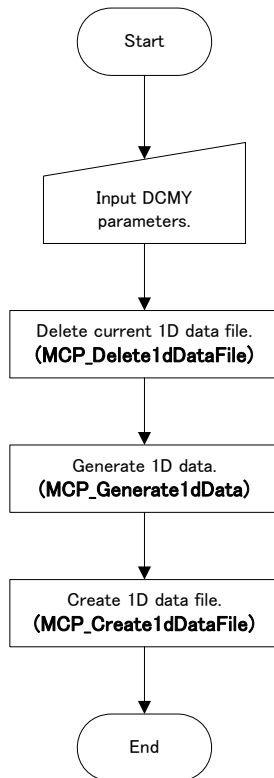
The DLL for the image-quality setup process is separated with the DLL for the printer control.

6. Color Correction Tool

In KIOSK system, we are investigating that Operator performs the simple color correction by the same operation as a mini-lab system when Operator feels the color is strange at the time of changing a new ink ribbon.

To resolve this, SDK has functions for simple color correction.

At the time to change ink



At the time of practical use

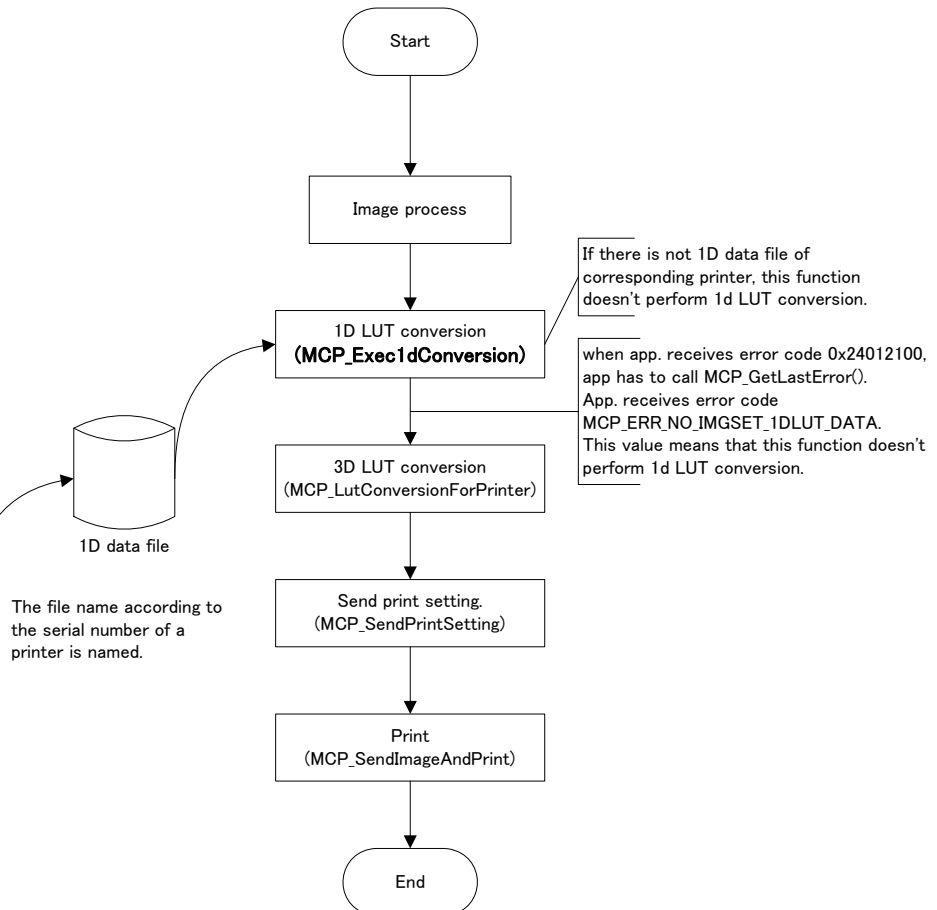


Figure 7: Control flow for Simple Color Correction at the time of change ink ribbon and the time of practical use.

7. SDK Control Procedure

In this section, the way to control ASK-1500/2000/2500/4000/4000A printer using the SDK is explained.

The function currently written here is an indispensable function.

It is necessary to surely call in this turn.

ASK-1500/2000/2500/4000/4000A have a memory for two pages for all paper sizes.

Therefore, application can send the next image to printer on printing.

This followings shows only a typical basic procedure. The printer can be controlled using other procedures.

The control methods are described also in the SDK sample programs. See these samples for your information.

7.1 Flow of Print Process for ASK-1500/2000/2500/4000/4000A

7.1.1 Print Flow

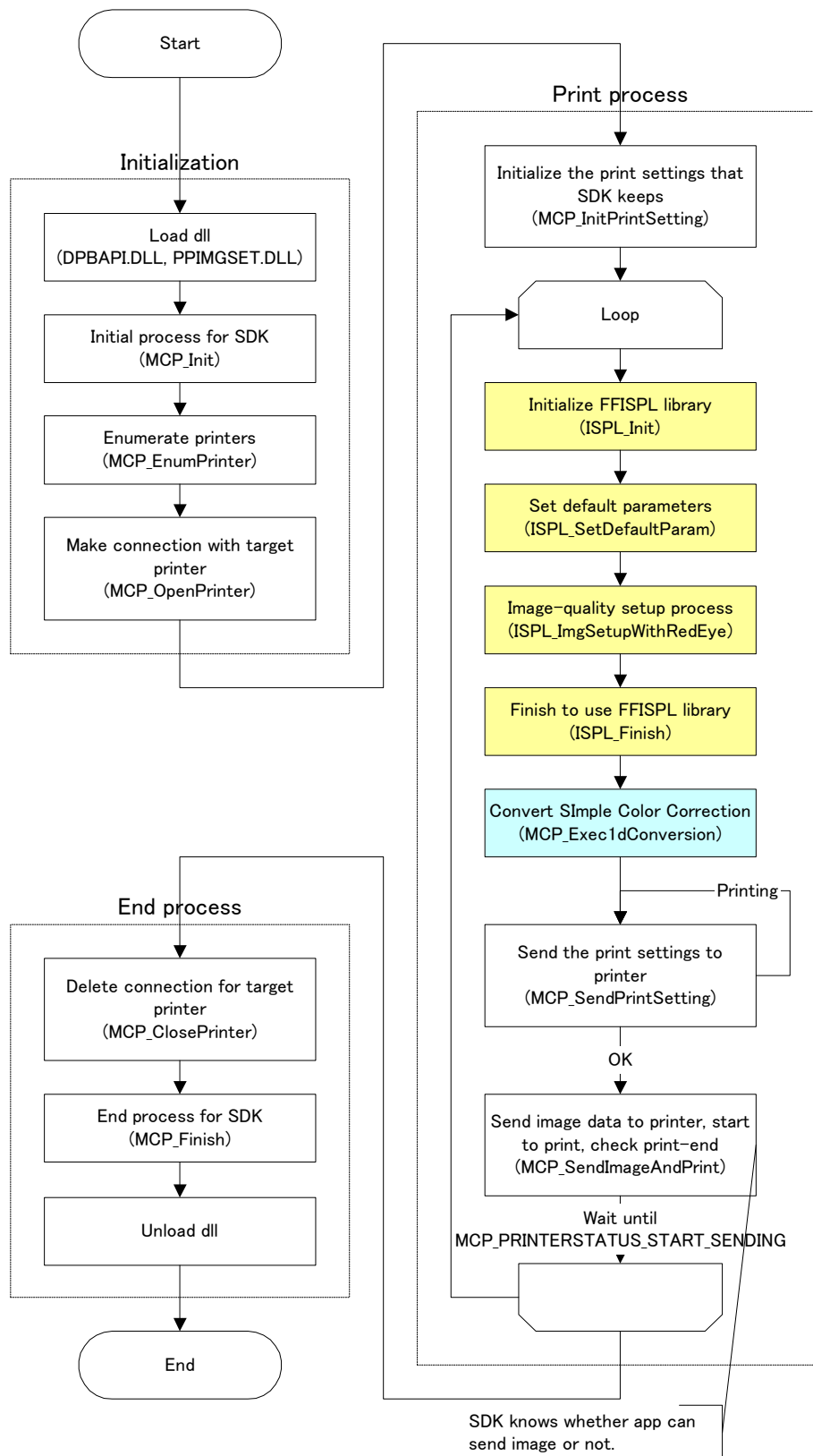


Figure 8: Process flow to print

Initialization process

1. Load the SDK dll (DPBAPI.DLL) and I2 dll (PPIMGSET.DLL).
2. Initialize the SDK. (**MCP_Init**)
3. Enumerate the printers. (**MCP_EnumPrinter**)
4. Open a connection with a target printer to be controlled. (**MCP_OpenPrinter**)

Printing process

5. Initialize the print setting information held in the SDK. (**MCP_InitPrintSetting**)
6. Prepare image data to be printed.
7. Initialize FFISPL library. (**ISPL_Init**)
8. Set the default parameters in FFISPL. (**ISPL_SetDefaultParam**)
9. Perform the image-quality setup process, red-eye removal process and conversion of color space. (**ISPL_ImgSetupWithRedEye**)
10. Finish to use FFISPL library. (**ISPL_Finish**)
11. Perform 1d LUT conversion for Simple Color Correction. (**MCP_Exec1dConversion**)
12. Send print setting information to a target printer. (**MCP_SendPrintSetting**)

If application receives **MCP_OK**, it will progress next.

If application receives the return code as follows, application has to send **MCP_GetLastError**.

*MCP_ERRF_P as flag

*MCP_PERR_PRINTER_BUSY as ErrInfo

*MCP_RECOVERY_WAIT_READY as Recovery

Then, if application receives **MCP_ERR_BUSY_PRINTING**, application has to wait until the current image is finished to print.

13. Send image data to a target printer, instruct to start printing, and monitor the progress of printing. (**MCP_SendImageAndPrint**)

Via the callback function, receive the status and work status to manage the printing status of images.

When callback function receives **MCP_PRINTERSTATUS_START_SENDING**, application can start to perform the process for the next image.

When callback function receives **MCP_PRINTERSTATUS_SEND_COMPLETE**, application can free the memory for image data.

End Process

14. Close a connection with a target printer being controlled. (**MCP_ClosePrinter**)
15. Exit the SDK. (**MCP_Finish**)
16. Release the SDK dll and I2 dll that has been loaded.

7.1.2 Printing Information on Printing

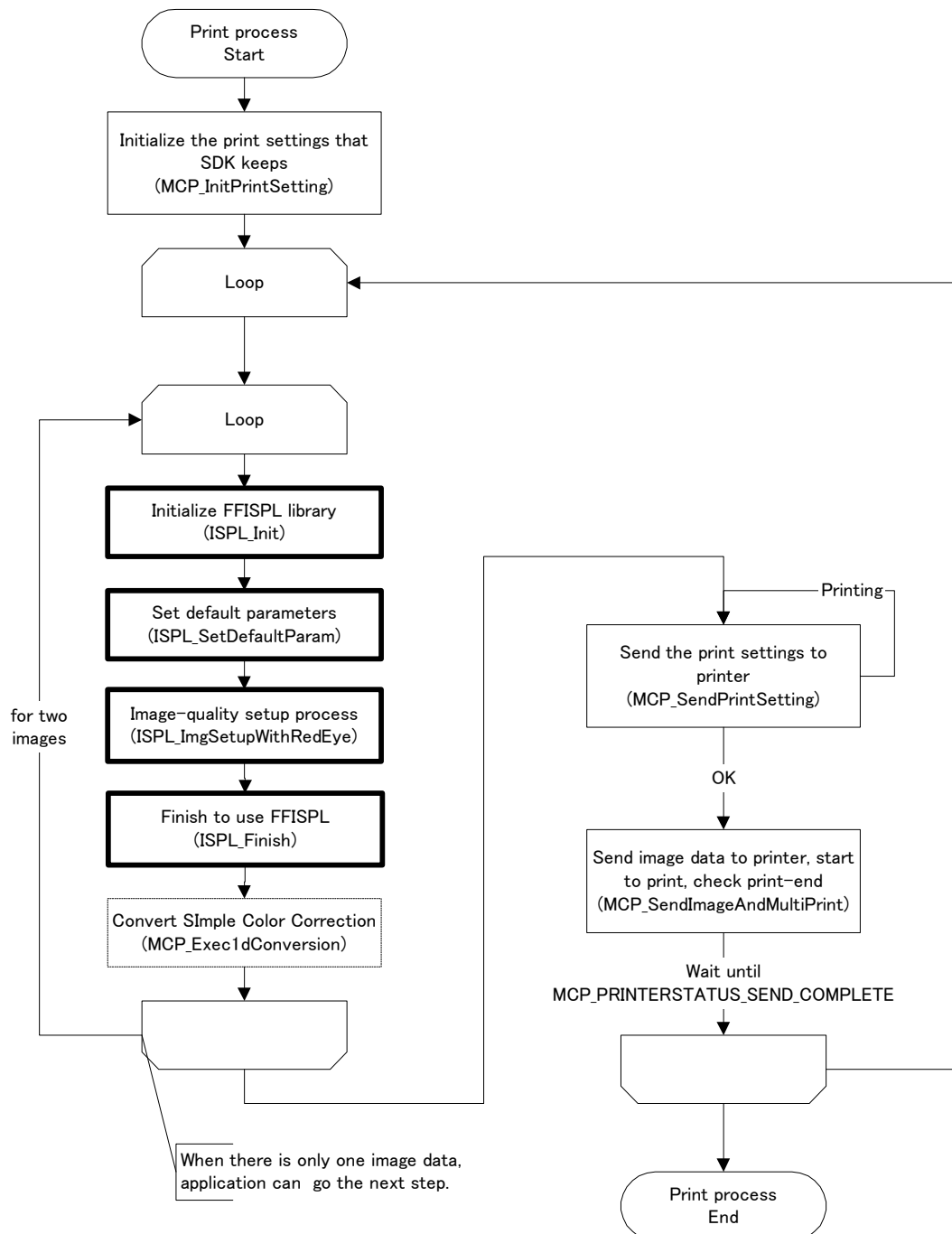
Status	bDetailStatus (MCP_PRINTERSTATUS_)	wJobNumber	wPrintedNumber	wPrintRequestNumber
(Call MCP_SendImageAndPrint for 1 st image.)		-	-	-
Start to send 1 st image.	START_SENDING	1	0	1
Complete to send 1 st image.	SEND_COMPLETE	1	0	1
Start to print 1 st image.	PRINTING	1	0	1
(Call MCP_SendImageAndPrint for 2 nd image)		-	-	-
Start to send 2 nd image.	START_SENDING	2	0	1
Complete to send 2 nd image.	SEND_COMPLETE	2	0	1
Finish to print 1 st image.	PRINTING	1	1	1
Start to print 2 nd image.	PRINTING	2	0	1
(Call MCP_SendImageAndPrint for 3 rd image.)		-	-	-
Start to send 3 rd image.	START_SENDING	3	0	2
Complete to send 3 rd image.	SEND_COMPLETE	3	0	2
Finish to print 2 nd image.	PRINTING	2	1	1
Start to print 3 rd image.	PRINTING	3	0	2
Finish to print 3 rd image for 1 st copy.	PRINTING	3	1	2
Finish to print 3 rd image for 2 nd copy	PRINTING	3	2	2
Ready	READY	0	0	0

7.2 Flow of Print Process for Multi Print Mode

ASK-1500/2000/2500 has the multi print mode that is printed two 4"x6" images on 6"x9" paper at the same time.

In this section, this multi print mode is explained.

7.2.1 Print Flow



Printing process

1. Initialize the print setting information held in the SDK. (**MCP_InitPrintSetting**)
2. Prepare image data to be printed.

3. Initialize FFISPL library. (**ISPL_Init**)
4. Set default parameters in FFISPL. (**ISPL_SetDefaultParam**)
5. Perform the image-quality setup process. (**ISPL_ImgSetupWithRedEye**)
6. Finish to use FFISPL library. (**ISPL_Finish**)
7. Perform 1d LUT conversion for Simple Color Correction. (**MCP_Exec1dConversion**)
8. Execute process 2-7 again to prepare two images to print.

When there is only one image data, application can go next step.

9. Send print setting information to a target printer. (**MCP_SendPrintSetting**)

If application receives **MCP_OK**, it will progress next.

If application receives the return code as follows, application has to send **MCP_GetLastError**.

*MCP_ERRF_P as flag

*MCP_PERR_PRINTER_BUSY as ErrInfo

*MCP_RECOVERY_WAIT_READY as Recovery

Then, if application receives **MCP_ERR_BUSY_PRINTING**, application has to wait until the current image is finished to print.

10. Send two image data to a target printer, instruct to start printing, and monitor the progress of printing. (**MCP_SendImageAndMultiPrint**)

Via the callback function, receive the status and work status to manage the printing status of images.

When callback function receives **MCP_PRINTERSTATUS_SEND_COMPLETE**, application can free the memory for image data.

7.2.2 Printing Information on Printing

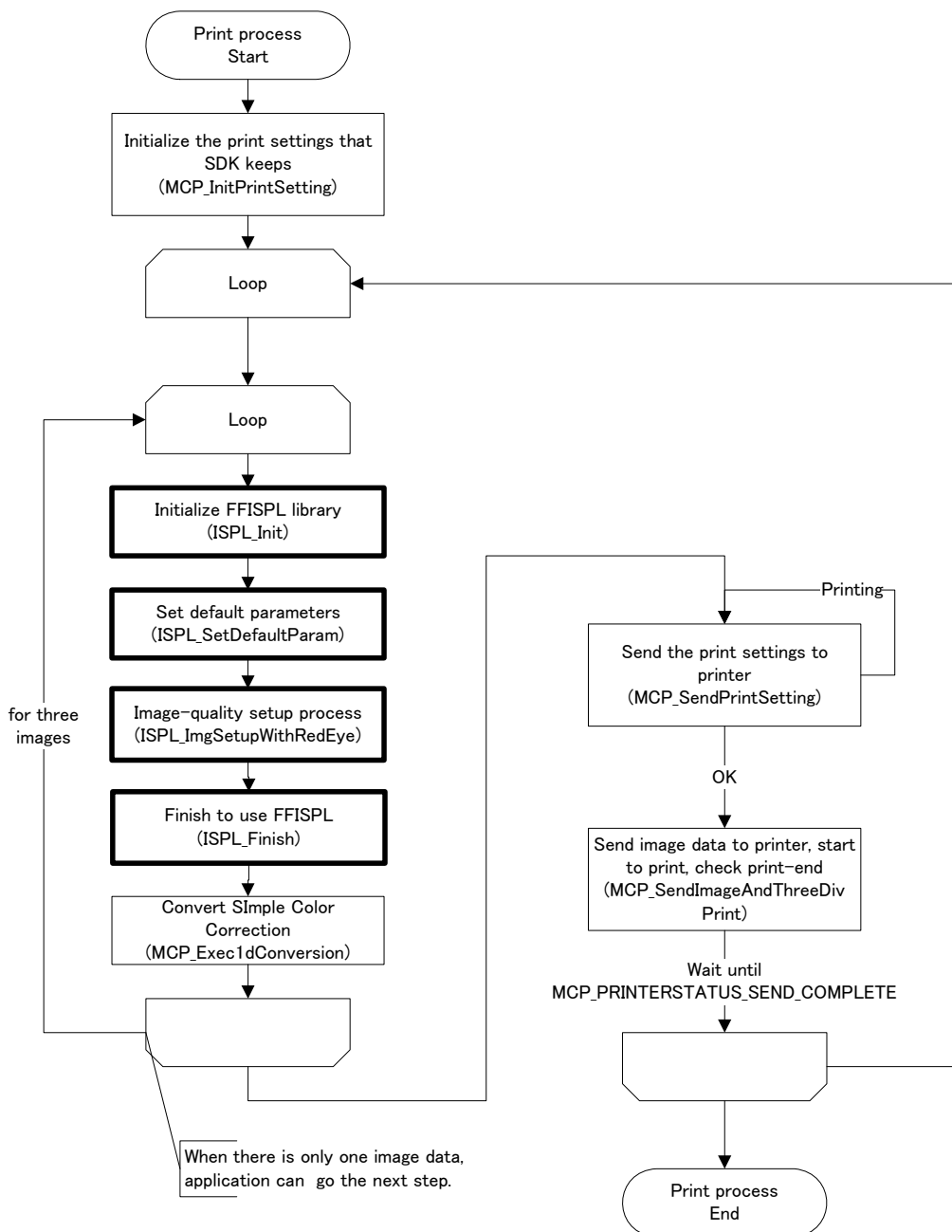
Status	bDetailStatus (MCP_PRINTERSTATUS_)	wJobNumber	wPrintedNumber	wPrintRequestNumber
(Prepare 1 st and 2 nd image.)		-	-	-
(Call MCP_SendImageAndMultiPrint for 1 st and 2 nd image.)		-	-	-
Complete to send 1 st and 2 nd image.	SEND_COMPLETE	1	0	1
Start to print 1 st and 2 nd image.	PRINTING	1	0	1
(Prepare 3 rd and 4 th image.)		-	-	-
(Call MCP_SendImageAndMultiPrint for 3 rd and 4 th image.)		-	-	-
Complete to send 3 rd and 4 th image.	SEND_COMPLETE	2	0	1
Finish to print 1 st and 2 nd image.	PRINTING	1	1	1
Start to print 3 rd and 4 th image.	PRINTING	2	0	1
Finish to print 3 rd and 4 th image.	PRINTING	2	1	1
Ready	READY	0	0	0

7.3 Flow of Print Process for Three Division Print Mode

ASK-4000 has the multi print mode that is printed three 8"x4" images on 8"x12" paper at the same time.

In this section, this three division print mode is explained.

7.3.1 Print Flow



Printing process

1. Initialize the print setting information held in the SDK. (**MCP_InitPrintSetting**)
2. Prepare image data to be printed.
3. Initialize FFISPL library. (**ISPL_Init**)
4. Set default parameters in FFISPL. (**ISPL_SetDefaultParam**)

5. Perform the image-quality setup process. (**ISPL_ImgSetupWithRedEye**)
6. Finish to use FFISPL library. (**ISPL_Finish**)
7. Perform 1d LUT conversion for Simple Color Correction. (**MCP_Exec1dConversion**)
8. Execute process 2-7 again to prepare two images to print.

When there is only one image data, application can go next step.

9. Send print setting information to a target printer. (**MCP_SendPrintSetting**)

If application receives **MCP_OK**, it will progress next.

If application receives the return code as follows, application has to send **MCP_GetLastError**.

*MCP_ERRF_P as flag

*MCP_PERR_PRINTER_BUSY as ErrInfo

*MCP_RECOVERY_WAIT_READY as Recovery

Then, if application receives **MCP_ERR_BUSY_PRINTING**, application has to wait until the current image is finished to print.

10. Send three image data to a target printer, instruct to start printing, and monitor the progress of printing. (**MCP_SendImageAndThreeDivPrint**)

Via the callback function, receive the status and work status to manage the printing status of images.

When callback function receives **MCP_PRINTERSTATUS_SEND_COMPLETE**, application can free the memory for image data.

7.3.2 Printing Information on Printing

Status	bDetailStatus (MCP_PRINTERSTATUS_)	wJobNumber	wPrintedNumber	wPrintRequestNumber
(Prepare 1 st , 2 nd and 3 rd image.)		-	-	-
(Call MCP_SendImageAndMultiPrint for 1 st , 2 nd and 3 rd image.)		-	-	-
Complete to send 1 st , 2 nd and 3 rd image.	SEND_COMPLETE	1	0	1
Start to print 1 st , 2 nd and 3 rd image.	PRINTING	1	0	1
(Prepare 4 th , 5 th and 6 th image.)		-	-	-
(Call MCP_SendImageAndMultiPrint for 4 th , 5 th and 6 th image.)		-	-	-
Complete to send 4 th , 5 th and 6 th image.	SEND_COMPLETE	2	0	1
Finish to print 1 st , 2 nd and 3 rd image.	PRINTING	1	1	1
Start to print 4 th , 5 th and 6 th image.	PRINTING	2	0	1
Finish to print 4 th , 5 th and 6 th image.	PRINTING	2	1	1
Ready	READY	0	0	0

7.4 How to Control Two ASK-1500/2000/2500

This section is explained how to control two ASK-1500/2000/2500 printers.

Control flow is follows.

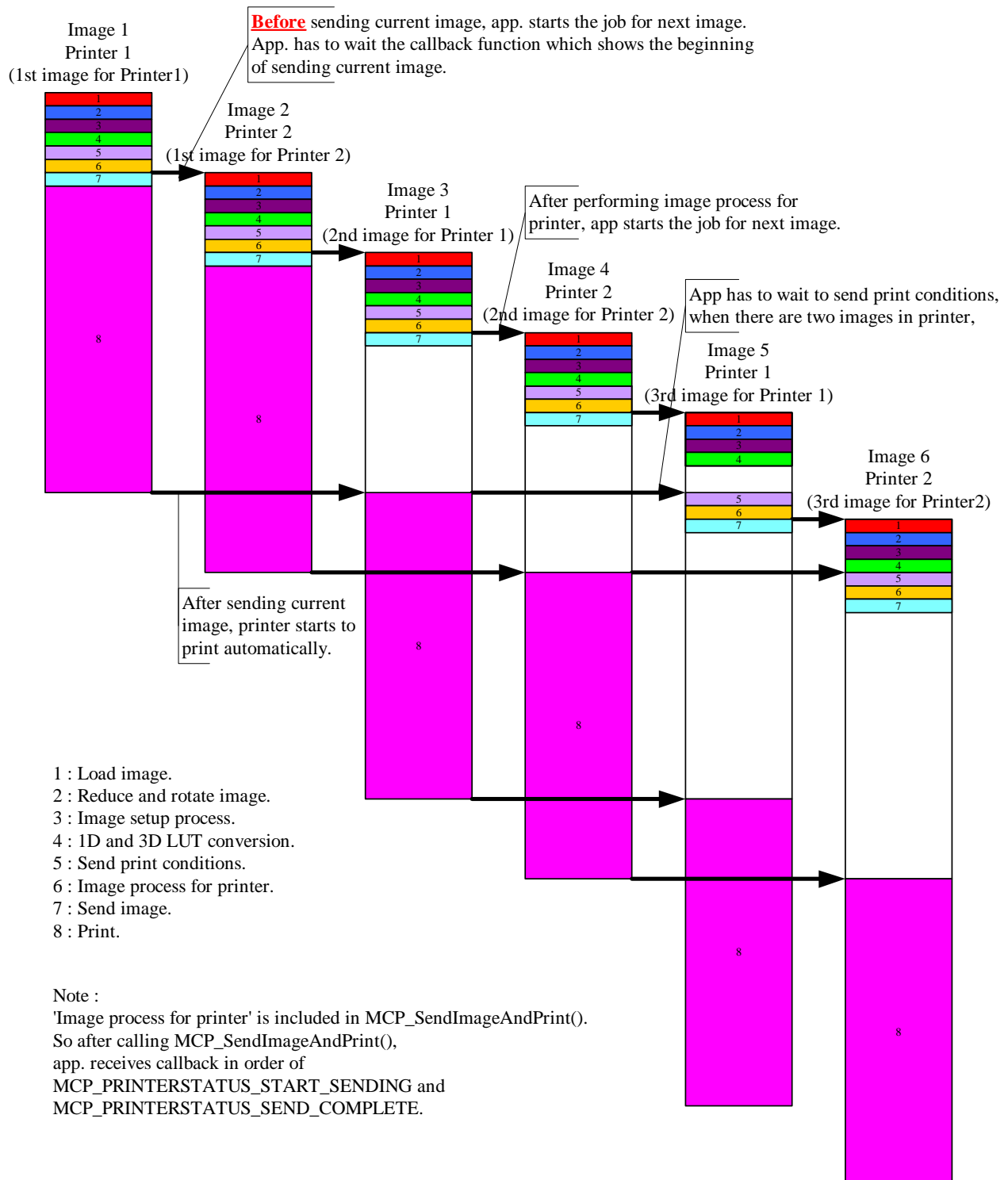


Figure 9: Time chart to print

As shown in above figure, when app. receives MCP_PRINTERSTATUS_START_SENDING as callback, app. starts to perform the processing for the next image.

8. SDK Error Code

8.1 API Function Error Code Details

b00 - b07	b08 - b15	b16 - b23	b24 - b28	b29 b31 b30
Reserved	Recovery	ErrInfo	Flag	C 0

An error flag (Flag) is seen first.

This error flag has a following meaning and error information (ErrInfo) is classified, respectively.

Here, the contents of error information are explained for every error flag.

Furthermore, recovery information (Recovery) can be acquired and the solution of an error can be distributed referring to this recovery information.

Moreover, when you acquire error information detailed information, call **MCP_GetLastError**.

C: Customer error flag (1)

Flag: Error flag

0x00	MCP_ERRF_OK	No error
0x01	MCP_ERRF_P	Printer condition error
0x02	MCP_ERRF_H	Printer hardware error
0x03	MCP_ERRF_I	Communication error
0x04	MCP_ERRF_S	Software error
0x12	MCP_ERRF_H_CONTINUE	Printer hardware error (SDK continues to run the printing process)

ErrInfo: Error information

Recovery: Recovery information

0x01	MCP_RECOVERY_IGNORE	Can be ignored.
0x02	MCP_RECOVERY_CMD_RETRY	Command retry is required.
0x03	MCP_RECOVERY_WAIT_READY	Wait for a READY status.
0x05	MCP_RECOVERY_CLOSE_PRINTER	Close a connection and perform retry.
0x11	MCP_RECOVERY_OPERATOR_RESET	Reset by operator is required.
0x12	MCP_RECOVERY_REBOOT_PRINTER	Restart a printer.
0x13	MCP_RECOVERY_REBOOT_PC	Restart a PC.
0x14	MCP_RECOVERY_EXHAUST_PARTS	Replace parts.
0x21	MCP_RECOVERY_CHECK_ENV	Check the environment (File existence)
0x23	MCP_RECOVERY_CHECK_DEVICE	Check the local connection.

0x29	MCP_RECOVERY_CHECK_PROGRAM	Check the program.
0x3F	MCP_RECOVERY_UNKNOWN	Unknown

The SDK provides the following macros to facilitate programming.

```
#define MCP_ERRFLAG(dwRet)      (BYTE)((dwRet & 0x1F000000) >> 24)
#define MCP_ERRINFO(dwRet)     (BYTE)((dwRet & 0x00FF0000) >> 16)
#define MCP_RECOVERY(dwRet)    (BYTE)((dwRet & 0x0000FF00) >> 8)
```

Below, the contents of error information are explained for every error flag.

8.1.1 Printer Condition Error (Flag:0x01 MCP_ERRF_P)

ErrInfo:

0x01	MCP_PERR_PRINTER_BUSY	Printer BUSY
0x02	MCP_PERR_NO_PRINTER	No printer
0x03	MCP_PERR_PRINTER_RESERVED	The printer is reserved by someone else.
0x04	MCP_PERR_INVALID_CONNECT	The connection is invalid.
0x3F	MCP_PERR_UNKNOWN	Unknown error

8.1.2 Printer Hardware Error (Flag:0x02 MCP_ERRF_H)

ErrInfo:

0x01	MCP_HERR_H	Hardware error
0x02	MCP_HERR_J	Paper jamming error
0x03	MCP_HERR_U	Caution
0x04	MCP_HERR_C	Communication error
0x11	MCP_HERR_NO_PAPER	Out of paper
0x12	MCP_HERR_PARAMETER	Parameter error
0x3F	MCP_HERR_UNKNOWN	Unknown error

8.1.3 Communication Error (Flag:0x03 MCP_ERRF_I)

ErrInfo:

0x01	MCP_IERR_COMM_DEVICE	Local interface error
0x3F	MCP_IERR_UNKNOWN	Unknown error

8.1.4 Software Error (Flag:0x04 MCP_ERRF_S)

ErrInfo:

0x01	MCP_SERR_USE_SDK	SDK usage error
0x02	MCP_SERR_PC_RESOURCE	PC resource error
0x3F	MCP_SERR_UNKNOWN	Unknown error

8.1.5 Printer Hardware Error (Flag:0x12 MCP_ERRF_H_CONTINUE)

When this error has occurred, SDK continues to run the printing process and notification for callback.

When this error has occurred, printer automatically prints again after error is solved.

ErrInfo:

0x03	MCP_HERR_U	Caution
0x11	MCP_HERR_NO_PAPER	Out of paper

8.2 MCP_GetLastError Code Details

b31 · b24	b23 · b00
Flag	Error code

The return value of **MCP_GetLastError** has such a meaning for every bit. Please see an error flag (Flag) first.

This error flag has a following meaning and the detailed error code (Error code) is classified, respectively.

Flag:

Error flag

0x00	MCP_ERRF_OK	No error
0x01	MCP_ERRF_P	Printer condition error
0x02	MCP_ERRF_H	Printer hardware error (Non-recoverable)
0x12	MCP_ERRF_H_CONTINUE	Printer hardware error (Recoverable)
0x03	MCP_ERRF_I	Communication error
0x04	MCP_ERRF_S	Software error

Below, the contents of the detailed error code are explained for every error flag.

8.2.1 Printer Condition Error (Flag:0x01 MCP_ERRF_P)

Error code:

0x01 01 00 01	MCP_ERR_BUSY_INITIALIZING	Busy because initialization is in progress
0x01 01 00 03	MCP_ERR_BUSY_PRINTING	Busy because printing is in progress
0x01 01 00 F1	MCP_ERR_BUSY_FEEDANDCUT	Busy because feeding or cutting
0x01 01 00 FF	MCP_ERR_BUSY_UNKNOWN	Busy because of an unknown reason
0x01 02 00 01	MCP_ERR_NO_DEVICE	Printer not connected

8.2.2 Printer Hardware Error (Flag:0x02 MCP_ERRF_H)

When this error has occurred, it must restart printer.

Error code:

0x02 00 00 0A	MCP_ERR_NO_PAPER	No paper
0x02 00 00 0B	MCP_ERR_NO_INK	No ink ribbon
0x02 00 00 0C	MCP_ERR_PAPER_DOOR_OPEN	Paper door opens.
0x02 00 00 0D	MCP_ERR_INK_DOOR_OPEN	Ink ribbon door opens
0x02 00 00 0E	MCP_ERR_MEDIA_MISMATCH	Combination of paper and ink ribbon is illegal.
0x02 00 00 14	MCP_ERR_PAPER_JAM	Paper jamming.
0x02 00 00 15	MCP_ERR_INK_JAM	Ink ribbon jamming.
0x02 00 00 16	MCP_ERR_MECHA_ERROR	Mechanical error

8.2.3 Printer Hardware Error (Flag: 0x12 MCP_ERRF_H_CONTINUE)

When this error has occurred, printer automatically prints again after error is solved.

Error code:

0x12 00 00 0A	MCP_ERR_PRINTING_PAPER_END	Paper end in printing. (Recoverable)
0x12 00 00 0B	MCP_ERR_PRINTING_INK_END	Ink ribbon end in printing. (Recoverable)
0x12 00 00 0C	MCP_ERR_PRINTING_PAPER_DOOR_OPEN	Paper door opens in printing. (Recoverable)
0x12 00 00 0D	MCP_ERR_PRINTING_INK_DOOR_OPEN	Ink ribbon door opens in printing. (Recoverable)
0x12 00 00 0E	MCP_ERR_PRINTING_MEDIA_MISMATCH	Combination of paper and ink ribbon is illegal in printing. (Recoverable)

8.2.4 Communication Error (Flag:0x03 MCP_ERRF_I)

Error code:

0x03 01 01 01	MCP_ERR_USB_CONTROL	Error in USB control
---------------	---------------------	----------------------

8.2.5 Software Error (Flag:0x04 MCP_ERRF_S)

Error code:

0x04 01 00 01	MCP_ERR_NO_CONTROL	No method of printer control
0x04 01 01 52	MCP_ERR_NO_IMGSET_1DLUT_DATA	Failed to load 1D LUT data file in same folder of the SDK.
0x04 01 01 53	MCP_ERR_NO_IMGSET_3DLUT_DATA	Failed to load 3D LUT data file in same folder of the SDK.
0x04 01 02 01	MCP_ERR_NO_INIT	Not called MCP_Init().
0x04 01 02 02	MCP_ERR_STILL_OPEN	Not opened printer.
0x04 01 02 03	MCP_ERR_ALREADY_OPEN	Printer has already opened.
0x04 01 02 04	MCP_ERR_MAX_OPEN	More connections opened than the maximum number.
0x04 01 02 05	MCP_ERR_NO_OPEN	Printer control attempted in other than STATE3.
0x04 01 03 01	MCP_ERR_INVALID_HPRINTER	Invalid printer handle.
0x04 01 03 02	MCP_ERR_INVALID_PARAMETER	Invalid parameter
0x04 01 03 03	MCP_ERR_IMAGE_OUT_OF_PAPER	Too large image size.
0x04 01 03 04	MCP_ERR_NOTSUPPORT_IMAGE_FMT	Unsupported image format.
0x04 01 03 05	MCP_ERR_ALREADY_START_THREAD	Thread already started for the printer.
0x04 01 03 06	MCP_ERR_UNMATCH_IMAGESIZE	Invalid image size.
0x04 02 00 01	MCP_ERR_MEMORY_ALLOC	Memory allocation error.
0x04 02 00 02	MCP_ERR_FAIL_START_THREAD	Thread start error.

8.3 Error Recovery

When recoverable error has occurred in printing, printer or application performs the error recovery procedure.

The SDK has two modes as the error recovery procedure.

Printer (AutoRecoveryFlag : TRUE)

When recoverable error has occurred in printing, Printer automatically prints the images in printer memory after recovery.

Application should send the next image.

When using this mode, call MCP_SetAutoRecoveryFlag(TRUE).

Application (AutoRecoveryFlag : FALSE...Default)

When recoverable error has occurred in printing, the SDK removes the images in printer memory at this time.

Application should re-send these images.

When using this mode, don't call MCP_SetAutoRecoveryFlag(TRUE).

8.3.1 Printer

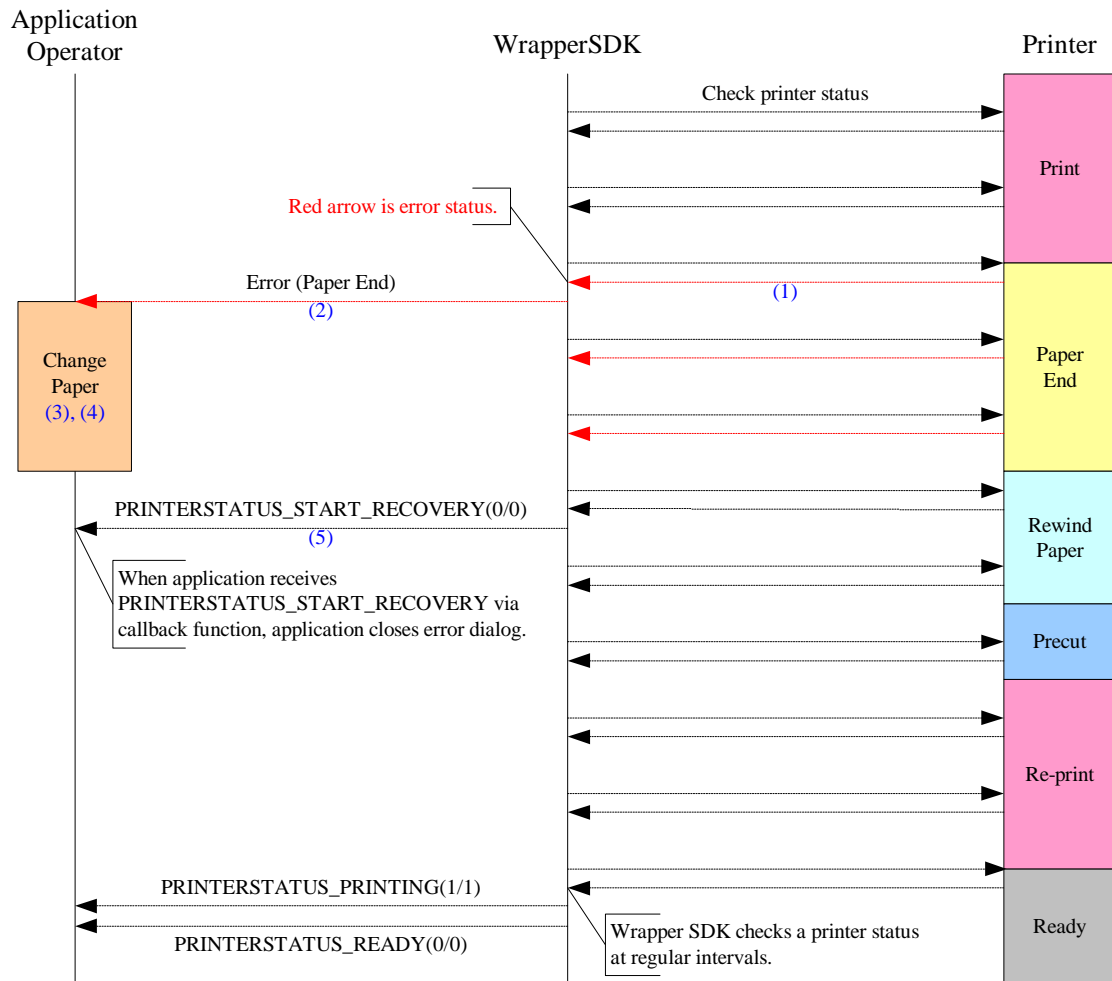
When recoverable error has occurred in printing, Printer cuts paper and prints again after recoverable error is solved.

In this section, the recovery process for recoverable error is explained.

<Example>

- (1) Paper end error has occurred.
- (2) The SDK reports paper end error to application.
- (3) Application displays an error dialog to change paper.
- (4) Operator changes paper.
- (5) The SDK reports starting recovery process to application.
(MCP_PRINTERSTATUS_START_RECOVERY)
- (6) Application closes an error dialog and waits print end.

So application has to judge whether error is recoverable or non-recoverable.



8.3.2 Application

When recoverable error has occurred in printing, the SDK removes the images existed in printer memory. Application should re-send these images.

In this section, the recovery process for recoverable error is explained.

<Example>

- (1) Paper end error has occurred.
- (2) The SDK reports paper end error to application.
- (3) The SDK removes the images existed in printer memory.
- (4) Application displays an error dialog to change paper.
- (5) Operator changes paper.
- (6) Application closes an error dialog and waits printer ready.
- (7) Application re-sends the images that the SDK removes

8.4 How to Judge Recoverable or Non-recoverable Error

Application can judge recoverable or non-recoverable error by the following way.

- (1) Application receives error via callback function.
- (2) Application sends MCP_GetLastError(), and receives the detailed error code.
- (3) Application checks the detailed error code.
- (4) Application calls MCP_IS_ERRF_H_MACRO to judge recoverable or non-recoverable error.

When non-recoverable error has occurred, printer needs to restart.

8.5 How to Get Error Code

When application gets the error code via SDK functions and callback functions, application can get the error information by the following way.

- (1) Application calls MCP_GetLastError(), and receives the detailed error code.
- (2) Application calls MCP_GetPrinterError(), and receives the string of Printer LCD error and the detailed printer status.
- (3) Application shows the error message according to the detailed error code, the string of Printer LCD error and the detailed printer status.

Notes

- (1) If the detailed error code is not printer error, the string of Printer LCD error is NULL.
- (2) If printer is ASK-1500, the string of Printer LCD error is NULL.

Example format

The detailed error code:	0x02010203
The string of Printer LCD error:	W21
The detailed printer status:	0x00123456

Error message is as follows.

0x02010203 Paper jamming (W21 – 0x00123456)

Please refer “ERRORCODELIST.XLS” attached in SDK.