

DP-DS620/DP-DS620 (A)
DP-DS820/DP-DS820 (A)

Panorama Processing API
(PanoramaProc.dll, PanoramaProc64.dll)

Function Reference

Ver. 1.10

Revision History Chart		First Edition: Dec 25,2015			Page 1/1	
Revised Item	Type of Revision	Revision No.	Document No.	Revision Date	Approved	Designer
	• New Release	1.0.0.0	1.00	Dec 25,2015		
1,2,4,11,12,14,16 6-10	<ul style="list-style-type: none"> • Added the API to Acquire the density rate • Added the API to process the density of the image data • Added the structure of the density processing • Stated clearly the input/output relationship of the arguments in the each API 	1.0.1.0	1.01	Feb 29,2016		
3 4,15,16	<ul style="list-style-type: none"> • DP-DS820/DP-DS820 (A) added • Changed the definition file for VB 	1.1.0.0	1.10	May 31,2016		

Contents

Introduction	3
API function	4
DLL Initialize	5
Image Data Segmentation	6
Image Overlap Processing (for left-side overlap image)	7
Image Overlap Processing (for right-side overlap image)	8
Image Overlap Processing (for both-side overlap image)	9
Environmental correction settings	10
Acquires the density rate	11
Process the density of the image data	12
Panorama process example	13
Color pattern creation example	14
Error Code List	15
Structure list	16

Introduction



The copyrights for this document are the property of the holders of rights. Reproduction of any or all of the contents is prohibited.



The contents of this document are subject to change without prior notice.



Microsoft, Windows, and .Net Framework are trademarks or registered trademarks of Microsoft Corporation in the USA and other countries.



You cannot resale/use the DP-DS620 and **the DP-DS820** without destination number (A) in the U.S.

Application Scope

This document is a command reference manual for the DP-DS620/DP-DS620 (A) and **the DP-DS820/DP-DS820 (A)** Panorama Processing API.

Supported Operating Systems and Operating Environment

This API runs on Windows XP, Windows Vista, Windows 7, Windows 8 and Windows 10.

API function

A table of the API functions is shown below.

■Notes

- If you are using C language, please include " PanoDef.h ", and add PanoramaProc.lib to the link library.
(For 64bitOS, use " PanoramaProc64.lib ")

When running this, put in " PanoramaProc.dll "(for 32BitOS) as a reference pass. (For 64bitOS, use " PanoramaProc64.dll ")

- When using VB.Net, since the pointer cannot be given directly from VB to the DLL function argument, a wrapper function for VB.Net with the same name as the DLL has been prepared.

The definition is written in the ClassPanoramaCommon.vb file and ClassPanoramaModule_x86/x64.vb file, so add the file to the VB project to use them.

Both ClassPanoramaModule_x86/x64.vb are defined such that they can be used and switched between 32⇔64 Bit OS environments.

For the 32 bit OS, make an instance of the ClassPanoramaModule_x86 class of the ClassPanoramaModule_x86.vb, and for the 64 bit OS, make an instance of the ClassPanoramaModule_x64 class of the ClassPanoramaModule_x64.vb.

Function	API function name	Point of notice
DLL Initialize	PP_Reset()	for pint sample application
Image Data Segmentation	PP_Divide()	for pint sample application
Image Overlap Processing (for left-side overlap image)	PP_ProcessLhs()	for pint sample application
Image Overlap Processing (for right-side overlap image)	PP_ProcessRhs()	for pint sample application
Image Overlap Processing (for both-side overlap image)	PP_ProcessBoth()	for pint sample application
Environmental correction settings	PP_SetCorrectionParam()	for pint sample application
Acquires the density rate	PP_GetDutyFromLUT()	for color sample application
Process the density of the image data	PP_DutyYMC()	for color sample application

DLL Initialize

[Format]	long PP_Reset (void)
[Parameter]	None
[Return]	Successful: 0 Failure: Error Code (refer to the Error Code List for contents)
[Note]	Initializes API
[Sample Coding]	<div>< Visual C ></div> <pre>long ret; ret = PP_Reset(); if(ret < 0){ // error }</pre> <div>< VB.NET ></div> <pre>Dim ret As Integer ret = PP_Reset() If ret < 0 Then GoTo Error End If</pre>

Image Data Segmentation

[Format]	long PP_Divide (char *pImgDst0, char *pImgDst1, char *pImgDst2, char *pImgSrc, StructDivideParam *pParam);
[Parameter]	pImgDst0 : pImgDst1 : pImgDst2 : pImgSrc : pParam :	OUT 1 st output data buffer for segmented image data (*1) OUT 2 nd output data buffer for segmented image data (*1) OUT 3 rd output data buffer for segmented image data (*1,*2) IN Memory address of segmented data image's source data IN Segmentation parameter
[Return]	Successful: Failure:	0 Error Code (refer to the Error Code List for contents)
[Note]	<p>The specified image is segmented as per the contents designated in the Segmentation parameter. After that, the segmented data is output to the specified buffer.</p> <p>*1 : The user should obtain sufficient data size for the image size.</p> <p>*2 : When the segmentation designation is 2, set this to NULL.</p>	
[Sample Coding]	<div>< Visual C ></div> <pre>long ret; ret = PP_Divide(OutputImage1, OutputImage2, OutputImage3, InputImage, param); if(ret < 0){ // error }</pre> <div>< VB.NET ></div> <pre>Dim ret As Integer ret = PP_Divide (OutputImage1, OutputImage2, OutputImage3, InputImage, param) If ret < 0 Then GoTo Error End If</pre>	

Image Overlap Processing (for left-side overlap image)

[Format]	long PP_ProcessLhs (char * pImgData, WCHAR *iccTblFilename, WCHAR *lutFilename, unsigned long overlapWidth, unsigned long srcWidth, unsigned long srcHeight);
[Parameter]	pImgData : iccTblFilename : lutFilename : overlapWidth : srcWidth : srcHeight :	IN/OUT Processing image data IN Color conversion table file name IN LUT data file name IN Image overlap width [dot] IN Image data width [dot] IN Image data height [dot]
[Return]	Successful: Failure:	0 Error Code (refer to the Error Code List for contents)
[Note]	The specified color conversion and LUT files are loaded, and the specified image data for the Continuous Panoramic Prints left-side overlap is processed. The processed image data will be written over the source data in the buffer.	
[Sample Coding]	<div>< Visual C ></div> <pre>long ret; ret = PP_ProcessLhs (IOImage, iccTblFilename, lutFilename, overlapWidth, srcWidth, srcHeight); if(ret < 0){ // error }</pre> <div>< VB.NET ></div> <pre>Dim ret As Integer ret = PP_ProcessLhs (IOImage, iccTblFilename, lutFilename, overlapWidth, srcWidth, srcHeight) If ret < 0 Then GoTo Error End If</pre>	

Image Overlap Processing (for right-side overlap image)

[Format]	long PP_ProcessRhs (char * pImgData, WCHAR *iccTblFilename, WCHAR *lutFilename, unsigned long overlapWidth, unsigned long srcWidth, unsigned long srcHeight);
[Parameter]	pImgData : iccTblFilename : lutFilename : overlapWidth : srcWidth : srcHeight :	IN/OUT Processing image data IN Color conversion table file name IN LUT data file name IN Image overlap width [dot] IN Image data width [dot] IN Image data height [dot]
[Return]	Successful: Failure:	0 Error Code (refer to the Error Code List for contents)
[Note]	The specified color conversion and LUT files are loaded, and the specified image data for the Continuous Panoramic Prints right-side overlap is processed. The processed image data will be written over the source data in the buffer.	
[Sample Coding]	<div>< Visual C ></div> <pre>long ret; ret = PP_ProcessRhs (IOImage, iccTblFilename, lutFilename, overlapWidth, srcWidth, srcHeight); if(ret < 0){ // error }</pre> <div>< VB.NET ></div> <pre>Dim ret As Integer Dim i As Integer ret = PP_ProcessRhs (IOImage, iccTblFilename, lutFilename, overlapWidth, srcWidth, srcHeight) If ret < 0 Then GoTo Error End If</pre>	

Image Overlap Processing (for both-side overlap image)

[Format]	long PP_ ProcessBoth (char * pImgData, WCHAR *iccTblFilename, WCHAR *lutFilename, unsigned long overlapWidth, unsigned long srcWidth, unsigned long srcHeight);
[Parameter]	pImgData : iccTblFilename : lutFilename : overlapWidth : srcWidth : srcHeight :	IN/OUT IN IN IN IN IN Processing image data Color conversion table file name LUT data file name Image overlap width [dot] Image data width [dot] Image data height [dot]
[Return]	Successful: Failure:	0 Error Code (refer to the Error Code List for contents)
[Note]	The specified color conversion and LUT files are loaded, and the specified image data for the Continuous Panoramic Prints both-side overlaps is processed. The processed image data will be written over the source data in the buffer.	
[Sample Coding]	<div>< Visual C ></div> <pre>long ret; ret = PP_ProcessBoth (IOImage, iccTblFilename, lutFilename, overlapWidth, srcWidth, srcHeight); if(ret < 0){ // error }</pre> <div>< VB.NET ></div> <pre>Dim ret As Integer Dim i As Integer ret = PP_ProcessBoth (IOImage, iccTblFilename, lutFilename, overlapWidth, srcWidth, srcHeight) If ret < 0 Then GoTo Error End If</pre>	

Environmental correction settings

[Format]	long PP_ SetCorrectionParam (double temp, double humidity, WCHAR *envTblFilename);		
[Parameter]	temp :	IN	Environmental temperature [°C]
	humidity :	IN	Environmental humidity [%]
	envTblFilename :	IN	Environmental correction table file name
[Return]	Successful:	0	
	Failure:	Error Code (refer to the Error Code List for contents)	
[Note]	<p>This sets the Environmental Corrections for image processing Before calling up the following image processing API's, by calling up this API and setting the Environmental Correction, the Environmental Correction function will be enabled with the image processing API process.</p> <ul style="list-style-type: none">• PP_ProcessLhs()• PP_ProcessRhs()• PP_ProcessBoth() <p>The settings for the Environmental Correction function will remain until PP_Reset is called up.</p>		
[Sample Coding]	<div>< Visual C ></div> <pre>long ret; ret = PP_ SetCorrectionParam (temp, humidity, envTblFilename); if(ret < 0){ // error }</pre> <div>< VB.NET ></div> <pre>Dim ret As Integer Dim i As Integer ret = PP_ SetCorrectionParam (temp, humidity, envTblFilename) If ret < 0 Then GoTo Error End If</pre>		

Acquires the density rate

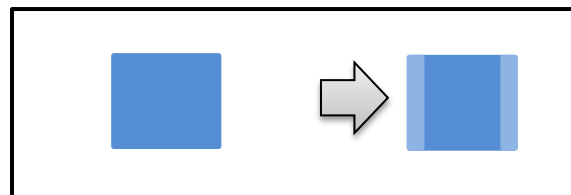
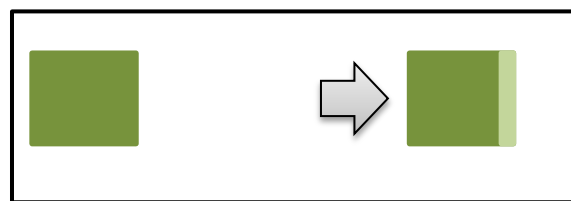
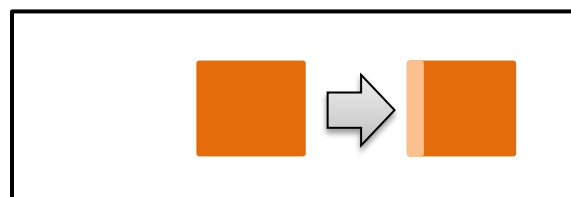
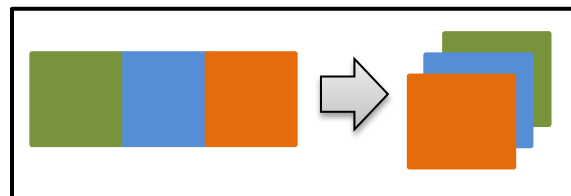
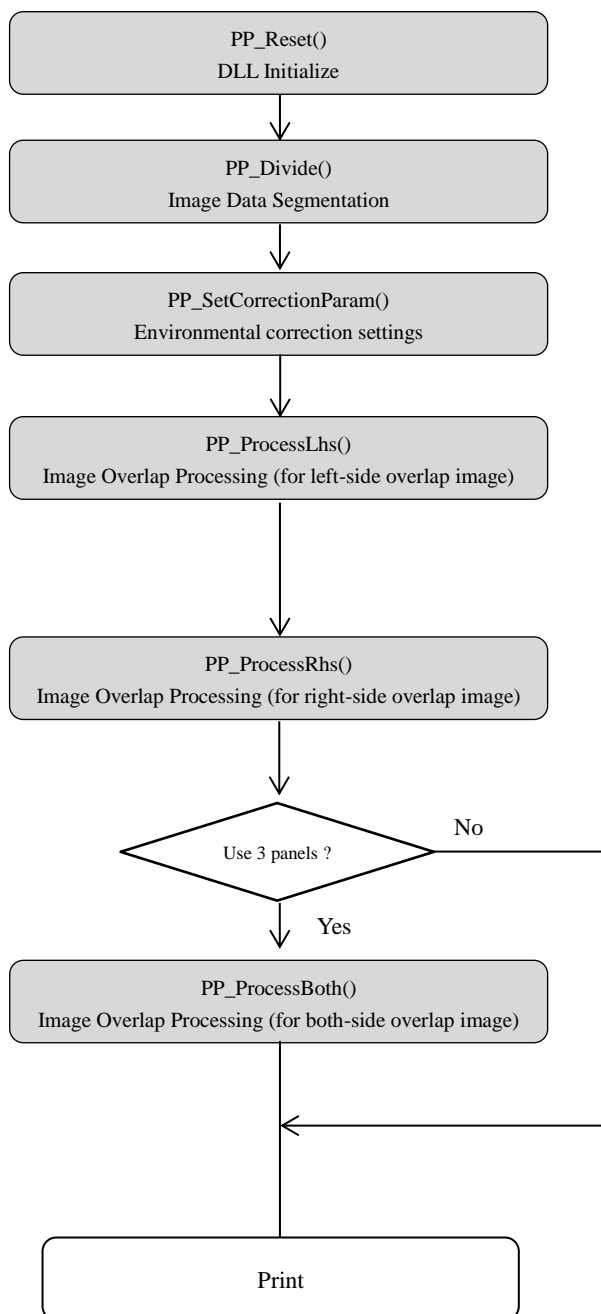
[Format]	long PP_GetDutyFromLUT(StructDutyOfLUTFile *param, WCHAR *lutFilename);		
[Parameter]	param :	IN/OUT	Storage structure for specified position and acquired density data
	lutFilename :	IN	LUT data file name
[Return]	Successful:	0	
	Failure:	Error Code (refer to the Error Code List for contents)	
[Note]	The specified LUT data file is loaded, and the density data for the specified position is acquired (Yellow, Magenta, and Cyan for 2 nd and 1 st prints). Here, the specified position will be matched to the scale defined in the LUT data file.		
[Sample Coding]	< Visual C > long ret; ret = PP_GetDutyFromLUT (param, lutFilename); if(ret < 0){ // error }		
	< VB.NET > Dim ret As Integer ret = PP_GetDutyFromLUT (param, lutFilename) If ret < 0 Then GoTo Error End If		

Process the density of the image data

[Format]	long PP_DutyYMC (BYTE *pImgData, StructImgDutyParam *param);		
[Parameter]	pImgData :	IN/OUT	Processing image data
	param :	IN	This parameter stores the image data size and processing density
[Return]	Successful:	0	
	Failure:	Error Code (refer to the Error Code List for contents)	
[Note]	With the specified parameters, the image data density is processed for each color: Yellow, Magenta, and Cyan. The processed image data will be written over the source data in the buffer.		
[Sample Coding]	< Visual C >		
	<pre>long ret; ret = PP_DutyYMC (pImgData, param); if(ret < 0){ // error }</pre>		
	< VB.NET >		
	<pre>Dim ret As Integer ret = PP_DutyYMC (pImgData, param) If ret < 0 Then GoTo Error End If</pre>		

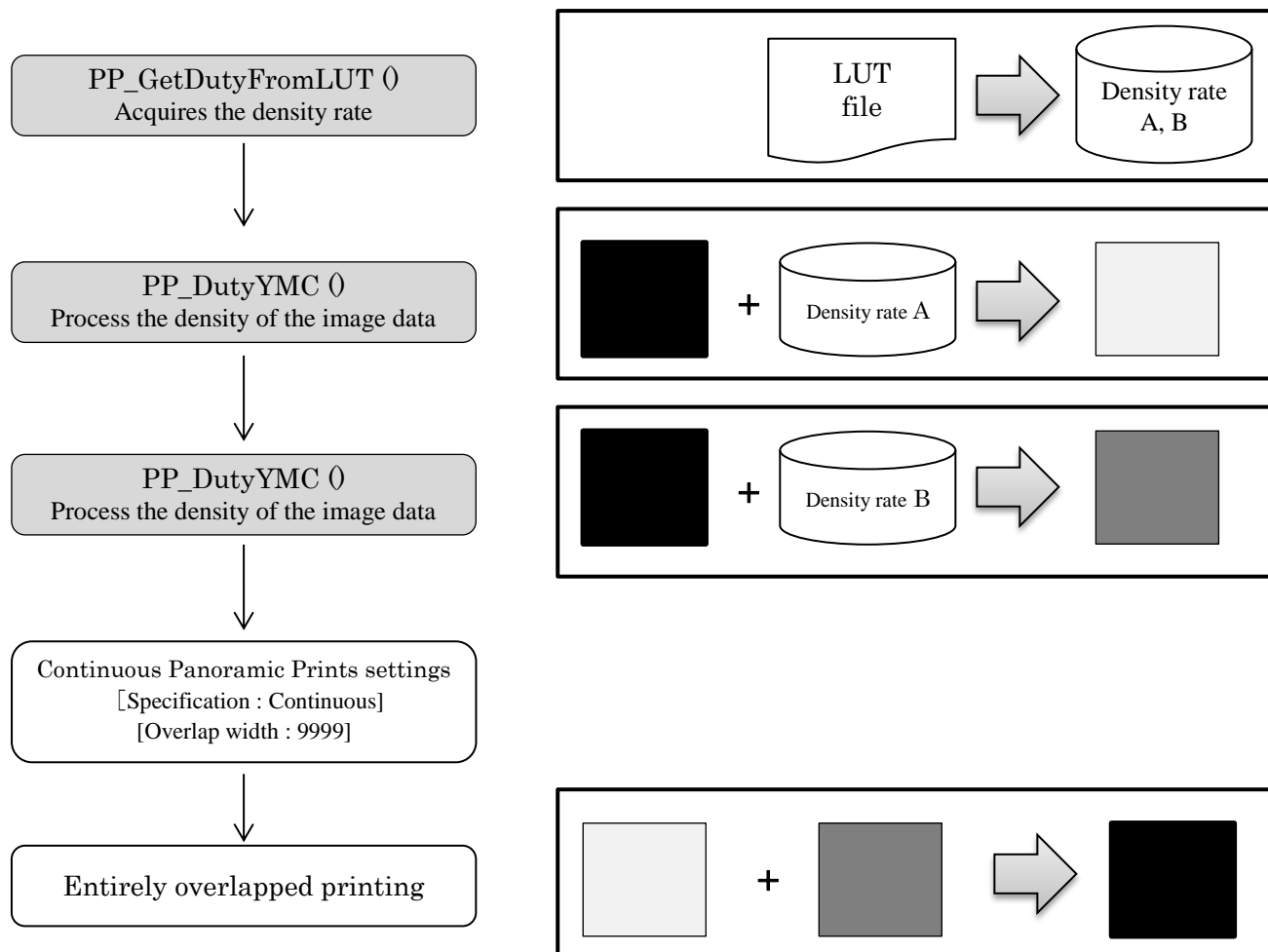
Panorama process example

A processing example using this API is shown below.



Color pattern creation example

A processing example using this API is shown below.



Error Code List

The error code setting at this API is shown below.

Also, the below definition name is defined in "PanoDef.h" and "ClassPanoramaCommon.vb" .

Value	Definition Name	Contents
0	NORMAL	Normal
-1	ERR_FILE_OPEN	File Open error
-2	ERR_NO_MEMORY	Insufficient memory
-3	ERR_PARAM	Parameter error
-4	ERR_OBJ	Incorrect condition
-5	ERR_NO_DATA	No data
-6	ERR_INTERNAL_PARAM	Internal process parameter error
-7	ERR_INTERNAL_PROC	Internal process error has occurred
-8	ERR_NO_SUPPORT	No support
-9	ERR_INTERNAL_NO_DUTY	No DUTY data in internal process
-10	ERR_INVALID_ENVIRONMENT	Invalid operating environment
-101	ERR_ICCTBLFILE_OPEN	Color conversion table file open error
-105	ERR_ICCTBLFILE_NODATA	No data in color conversion table file
-110	ERR_ICCTBLFILE_FORMAT	Color conversion table file format error
-201	ERR_ICCPROFILE_OPEN	Color conversion data file (ICC profile) open error
-206	ERR_ICCPROFILE_INIT	Error in color conversion initialize process
-207	ERR_ICCPROFILE_PROC	Error in color conversion process
-300	ERR_LUTFILE_OPEN	LUT File Open error
-305	ERR_LUTFILE_NODATA	No data in LUT File
-310	ERR_LUTFILE_FORMAT	LUT File Format error
-311	ERR_LUTFILE_ELM_TOOSMALL	Insufficient record count in LUT file (min. 4)
-400	ERR_CORRFILE_OPEN	Correction File Open error
-405	ERR_CORRFILE_NODATA	No correction file data
-410	ERR_CORRFILE_FORMAT	Correction file format error
-411	ERR_CORRFILE_DUPLICATE	Redundancy in the correction file data

Structure list

The structure to be defined by this API is shown below.

Also, the below structure name is defined in "PanoDef.h" and "ClassPanoramaCommon.vb".

Title	StructDivideParam		
Use	Image Segmentation Process parameter		
Member liste			
Name	Form	Size[byte]	Contents
numOfDiv	unsigned long	4	Number of segmentation
overlapWidth	unsigned long	4	Overlap width of segments [dot]
srcWidth	unsigned long	4	Source image width [dot]
srcHeight	unsigned long	4	Source image height [dot]
dstWidth	unsigned long	4	Image width after segmentation [dot]
dstHeight	unsigned long	4	Image height after segmentation [dot]

Title	StrutProcessParam		
Use	Continuous Panorama image processing parameters		
Member List			
Name	Form	Size [byte]	Contents
overlapWidth	unsigned long	4	Overlap width of image to be processed[dot]
srcWidth	unsigned long	4	Width of image to be processed [dot]
srcHeight	unsigned long	4	Height of image to be processed [dot]

Title	StructImgDutyParam		
Use	Image Density processing parameters		
Member List			
Name	Form	Size [byte]	Contents
srcWidth	unsigned long	4	Width of image to be processed [dot]
srcHeight	unsigned long	4	Height of image to be processed [dot]
dutyY	double	8	Yellow density(0=0%、 0.5=50%、 1.0=100%)
dutyM	double	8	Magenta density (")
dutyC	double	8	Cyan density (")

Title	StructDutyOfLUTFile		
Use	Acquires the LUT file designation location data		
Member List			
Name	Form	Size [byte]	Contents
pos	unsigned long	4	Designated horizontal position
scale	unsigned long	4	Designated scale (600 fixed)
resRhsDutyY	double	8	Acquired LUT density (2 nd print, Yellow) (Output)
resRhsDutyM	double	8	Acquired LUT density (2 nd print, Magenta) (Output)
resRhsDutyC	double	8	Acquired LUT density (2 nd print, Cyan) (Output)
resLhsDutyY	double	8	Acquired LUT density (1 st print, Yellow) (Output)
resLhsDutyM	double	8	Acquired LUT density (1 st print, Magenta) (Output)
resLhsDutyC	double	8	Acquired LUT density (1 st print, Cyan) (Output)