Windows SDK for Sanei Layout Designer2 User's Manual

June 09, 2021 Rev1.0.0

This manual provides information on the design guidelines concerning Sanei Layout Designer2 SDK that customers need to build Android applications.

⚠ SANEI ELECTRIC INC.

History of revision of this manual

Revision	Date	Description of revision
Rev1.0.0	June 09, 2021	Release of 1st edition (SDK Version1.00, EngineVer1.03)

Caution

・Windows SDK for Sanei LayoutDesigner2 is a copyrighted work of Sanei Electric Co., Ltd. (hereinafter referred to as
　Sanei Electric). Copyright and other rights concerning this product belong to Sanei Electric.

・Sanei Electric grants the user the right to use Windows SDK for Sanei LayoutDesigner2 (free to copy and distribute)
　for the purpose of using Sanei Electric products that are compatible with Windows SDK for Sanei LayoutDesigner2.

・Sanei Electric does not guarantee the absence of defects in the Windows SDK for Sanei LayoutDesigner2 and is not
　liable for any damages resulting from the use of the information contained in this manual.

・Sanei Electric shall never be liable under any circumstances for any direct or indirect loss or damage in connection with
　the use of Windows SDK for Sanei LayoutDesigner2.

・Users cannot directly or indirectly export all or part of the Windows SDK for Sanei LayoutDesigner2 without obtaining
　necessary permission from the Japanese government or the government of the relevant country.

1. Introduction

Windows SDK for Sanei Layout Designer2 is an assistive library that enables mutual editing by API based on the layout file (SOP file) of the Layout Designer2 software main body.
This class library file consists of the following library files (DLL files).

Class library file	LayoutDesigner2_SDK.dll
Barcode library	SaneiBarcodeLibrary.dll
2D Code Library	2d_Code_Dll.dll
Communication Library	Sk1_Api_dll.dll

1.1. Operating Environment

OS: Microsoft Windows 7 SP1
         Microsoft Windows 8
         Microsoft Windows 10

Supported printer models:
         SK1-21/31/41 Series
         SK1-21H/31H Series
         SK1-211/311 Series
         SD3-21/22 Series
         SK4-21/31 Series
         SM4-21/31 Series
         SM1-21 Series
         BLM-80 Series
         SM2-41 Series
         BL2-58 Series

Installation of .NET Framework 4.0 or higher is required to use this SDK.

1.2. Related Software
There is no related software in this version.

1.3. How to Implement the Class Library

(1) Applications that implement the Sanei Layout Designer2 SDK set the target framework to NET Framework 4.0 or higher.
You can view the target framework from the project properties.
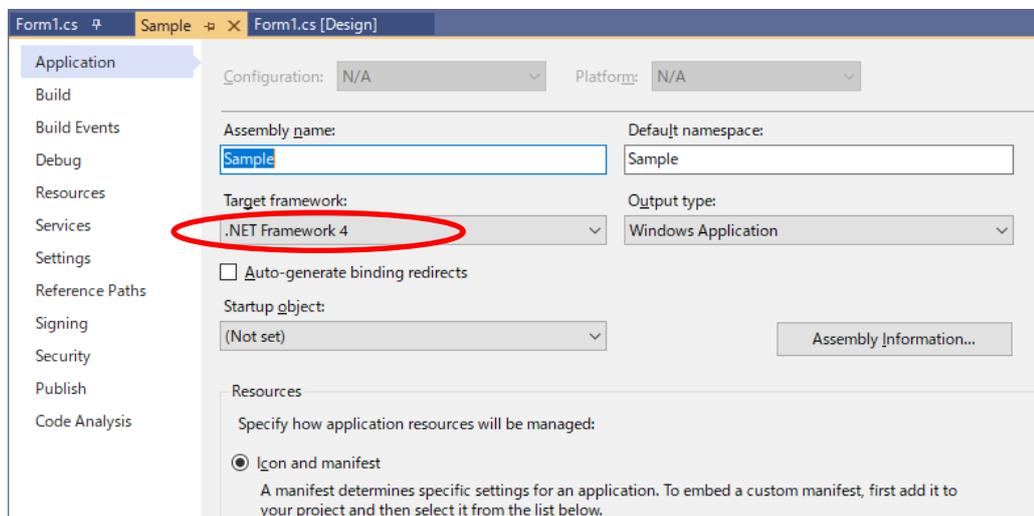
(2) Add the following DLL to the project folders of the application.

LayoutDesigner2_SDK.dll   Class library file
SaneiBarcodeLibrary.dll   Barcode library
2d_Code_Dll.dll   2D Code Library
Sk1_Api_dll.dll   Communication Library

Name	Date modified	Type	Size
Properties	5/31/2021 3:35 PM	File folder	
obj	5/31/2021 3:35 PM	File folder	
bin	5/31/2021 3:41 PM	File folder	
app.config	5/31/2021 3:46 PM	XML Configuratio...	1 KB
Program.cs	5/31/2021 3:35 PM	Visual C# Source f...	1 KB
Form1.Designer.cs	5/31/2021 3:40 PM	Visual C# Source f...	3 KB
Form1.cs	5/31/2021 3:46 PM	Visual C# Source f...	1 KB
Sample.csproj	5/31/2021 4:39 PM	Visual C# Project f...	4 KB
Form1.resx	5/31/2021 3:40 PM	Microsoft .NET M...	6 KB
Sk1_Api_dll.dll	6/30/2015 11:51 AM	Application exten...	59 KB
SaneiBarcodeLibrary.dll	5/18/2021 1:53 PM	Application exten...	2,239 KB
LayoutDesigner2_SDK.dll	5/21/2021 2:13 PM	Application exten...	2,440 KB
2d_Code_Dll.dll	9/12/2019 9:27 AM	Application exten...	202 KB

(3) Add the above four DLL files in "Add from Solution Explorer -> Existing Items".
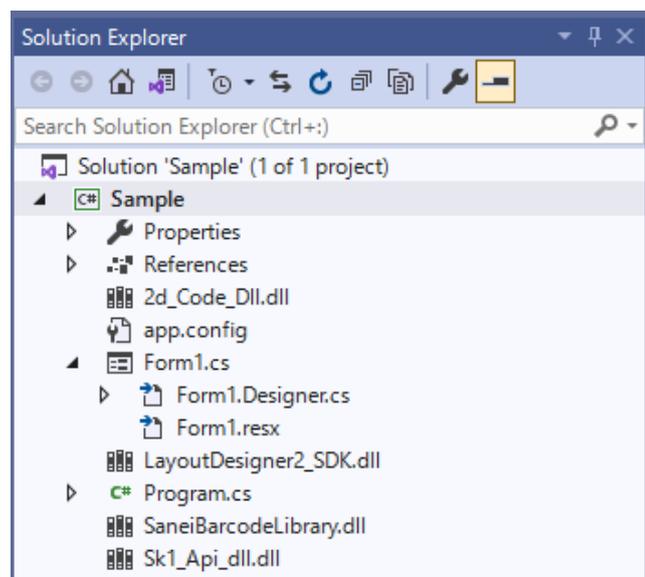
(4) Add LayoutDesigner2_SDK.dll in Add -> Ref from Solution Explorer.

Windows SDK for Sanei Layout Designer2
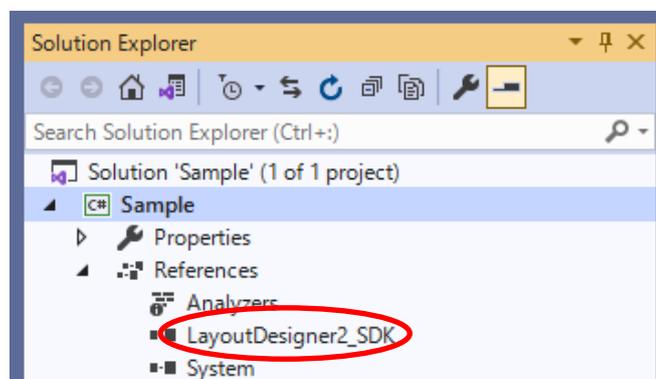
(5) Set the Platform target for the project to "x86".
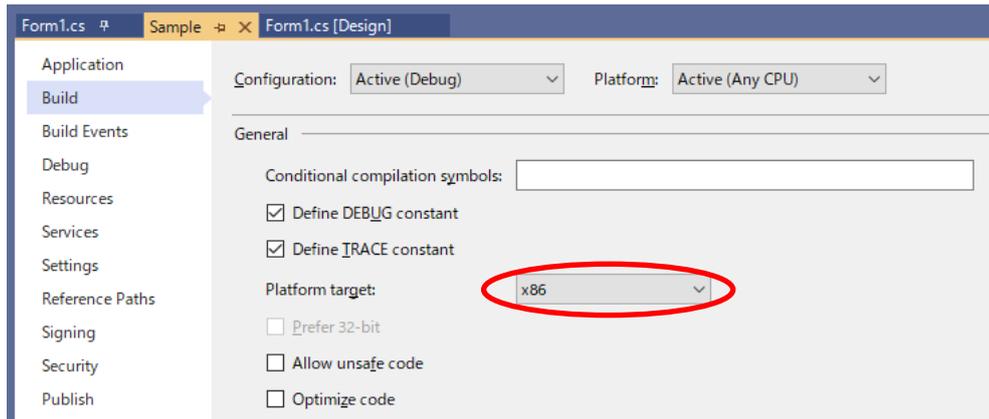
(6) Declare the class of SDK to the source code.
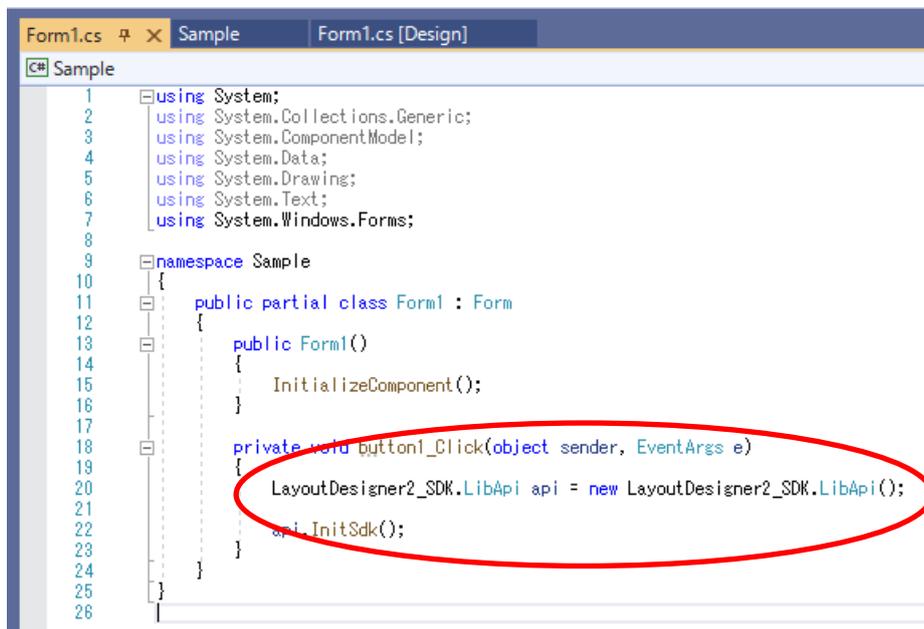
This completes the implementation and allows the LayoutDesigner2_SDK class to be used.

2. LayoutDesigner2_SDK class

The overall configuration using the LayoutDesigner2_SDK class and the process until printing are shown below.

● Overall configuration

```
┌─────────────────────────┐
│    User application     │
└─────────────────────────┘
              │
              ▼
┌─────────────────────────┐        ┌─────────────────────────┐
│      Class library      │◄───────│  Setting file (layout2.ini) │
│ (LayoutDesigner2_SDK Class) │    │   Layout file (SOP file)    │
└─────────────────────────┘        └─────────────────────────┘
              │
              ▼
┌─────────────────────────┐
│      Printer Driver     │
└─────────────────────────┘
              │
              ▼
┌─────────────────────────┐
│  Supported printer model │
└─────────────────────────┘
```
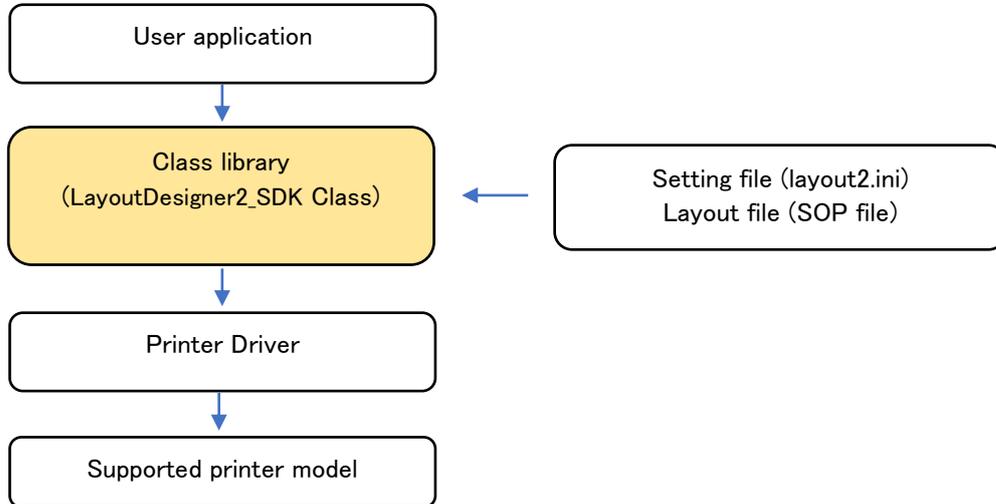
● Process to print the layout file

```
         ●
         │
         ▼
┌──────────────────────┐
│  Initializing the SDK │
└──────────────────────┘
         │
         ▼
┌──────────────────────┐
│  Loading a layout file │
└──────────────────────┘
         │
         ▼
┌──────────────────────────┐
│ Transmission of printing data │
└──────────────────────────┘
         │
         ▼
         ●
```
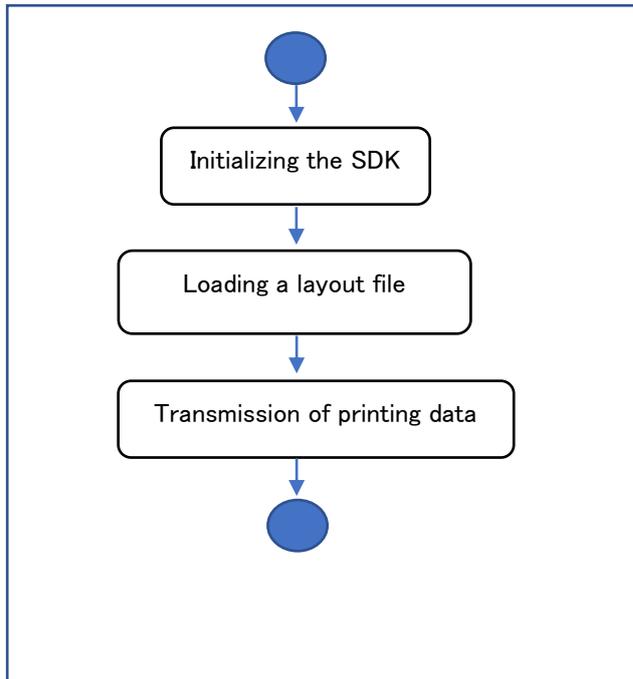
Sample code image:

```
//SDK initialization.
ret = api.InitSdk();
//Initialization failure?
if (ret == false)
{
      return;
}

//File load.
int file_size = 0;
ret = api.LoadSopFile(@"sample.sop", ref file_size);
//Load failed?
if (ret == false)
{
      return;
}

//Printing.
int copies = 1;
api.PrintSopFile(@"SANEI SK1-31S-UNI-JP", copies);
```

2.1. Global Area

The global regions that can be referenced are.

LibApi.ResultErrorCode   Error values by each API are stored.
For error values, refer to the error constant definition list in Chapter 4.

2.2. Setting file

The setting file refers to the Layout Designer2 preference file "layout2.ini".
The storage location of this file is located directly under the DLL folders.
In your app, place this configuration file in the same folders as LayoutDesigner2_SDK.dll.

Configuration of the preference file

image-print=1   Set whether to print image data.

      0:    Do not print anything other than the printer text.
      1:    Print everything.

auto-save=0   Sets whether the SOP file is automatically saved each time printing is performed.

      0:    The SOP file is not saved automatically.
      1:    The SOP file is automatically saved.

      For layout data using a serial number, setting this setting to "1" automatically saves the updated data of the serial number each time it is printed.

compress-print=1   Set whether to perform compression printing.

      0:    Compression printing is not performed.
      1:    Compression printing is performed.

3. API details

The function list for the LayoutDesigner2_SDK class is as follows:

Category	API	Function
Get version	GetApiVersion	Gets the version value of the SDK and internal engine.
Initialization	InitSdk	Initialize this class.
SOP file	NewSopFile	A new SOP file is created.
	LoadSopFile	Load the SOP file.
	SaveSopFile	Save the SOP file.
	PrintSopFile	Print the data of the current SOP file.
	PreviewSopFile	Creates a preview image of the current SOP file's data.
Serial number	InitSerial	Initialize the serial number.
	SetSerielNumber	Set the serial number.
Date/Time	SetDateTime	Set the date or time.
Object Editing	GetObjectList	Get the object list
	DelObjectData	Delete object data.
	GetTextData	Gets object data of text.
	SetTextData	Set object data for text.
	GetPTextData	Gets the object data of the printer text.
	SetPTextData	Set object data for printer text.
	GetImageData	The object data of the image is acquired.
	SetImageData	Set the object data of the image.
	GetLineData	Gets the object data of a straight line.
	SetLineData	Sets the object data of a straight line.
	GetRectData	Gets rectangular object data.
	SetRectData	Set rectangular object data.
	GetFillData	Gets fill object data.
	SetFillData	Set the fill object data.
	GetBarcode1dData	Obtains the object data of the barcode.
	SetBarcode1dData	Sets the object data of the barcode.
	GetBarcode2dData	Obtains the object data of the 2D code.
	SetBarcode2dData	Set the object data of the 2D code.
	GetMacroData	Gets the object data of the document macro.
	SetMacroData	Set the object data of the document macro.

3.1. GetApiVersion

Gets the version value of the SDK and internal engine.

Declaration:

bool　　　　LayoutDesigner2_SDK.LibApi.GetApiVersion(ref int sdk_ver, ref int eng_ver)

Argument:

int	sdk_ver	SDK version value
int	eng_ver	Internal engine version value

Return value:

true	Normal end
false	Error

Note:

A version value of 1.02 is returned as 102.

The version value of the internal engine returns the version of the layout designer 2 main unit.

3.2. InitSdk

Initialize this class.

Declaration:

bool　　　　LayoutDesigner.LibApi.InitSdk()

Argument:　　　　None

Return value:

true	Normal end
false	Error

3.3. NewSopFile

A new SOP file is created.

Declaration:

bool　　　　LayoutDesigner.LibApi.NewSopFile(int printer, int width, int height, int top, int left)

Argument:

int	printer	Printer model number
int	width	Print width (unit: dot pitch (1/8 mm))
int	height	Print height (unit: dot pitch (1/8 mm))
int	top	Top margin (unit is dot pitch (1/8 mm))
int	left	Left margin (unit is dot pitch (1/8 mm))

Return value:

true	Normal end
false	Error

Note:

For the printer model number, refer to the constant definition in Chapter 4.

3.4. LoadSopFile

Load the SOP file.

Declaration:

bool　　　LayoutDesigner.LibApi.LoadSopFile(string file, ref int size)

Argument:

string	file	File name of the SOP file
int	size	File size of the SOP file

Return value:

true	Normal end
false	Error

3.5. SaveSopFile

Save the SOP file.

Declaration:

bool　　　LayoutDesigner.LibApi.SaveSopFile(string file)

Argument:

string	file	File name of the SOP file

Return value:

true	Normal end
false	Error

3.6. PrintSopFile

Print the data of the current SOP file.

Declaration:

bool　　　LayoutDesigner.LibApi.PrintSopFile(string printer, int num)

Argument:

string	printer	Name of the printer driver
int	num	Number of copies printed

Return value:

true	Normal end
false	Error

3.7. PreviewSoPFile

Creates a preview image of the current SOP file's data.

Declaration:

bool   LayoutDesigner.LibApi.PreviewSoPFile(string file)

Argument:

| string | file | Name of the file of the image to be previewed. |
| | | Creating an image format in "PNG" format. |

Return value:

| true | Normal end |
| false | Error |

3.8. InitSerial

Initialize the serial number.

Declaration:

bool   LayoutDesigner.LibApi.InitSerial()

Argument: None

Return value:

| true | Normal end |
| false | Error |

3.9. SetSerialNumber

Set the serial number.

Declaration:

bool   LayoutDesigner.LibApi.SetSerialNumber(int SelectData, int SerialNumberType, int SerialIncDec,
                    int ObjectIncDec, int Counter, string SerialFormat, int CounterInit, int CounterMax)

Argument:

int	SelectData	Specifying object types
		0: Text system
		1: Barcode system
int	SerialNumberType	(Serial number) type specification
		0: Decimal number
int	SerialIncDec	(Serial number) Serial number increase/decrease specification
		0: Sequential number increase
		1: Sequential decrease
int	ObjectIncDec	Specifying the number of prints using the same serial number
int	Counter	To specify the number of increments/decrements
string	SerialFormat	Serial number format
		Write in To String method format of .NET Framework
int	CounterInit	To specify the initial count number
int	CounterMax	Specifying the maximum count number

Return value:

true	Normal end
false	Error

Note:

For the object type, type, and increase/decrease of sequential number, refer to the constant definition in Chapter 4.

3.10. SetDateTime

Set the date or time.

Declaration:

bool   LayoutDesigner.LibApi.SetDateTime(int DateTimeSelect, int FormatType, int Offset1, int Offset2)

Argument:

int	DateTimeSelect	Select date or time
		0: Date
		1: Time
int	FormatType	Specification of the format
		Date format number: 0 to 24
		Format number of the time: 0 to 19

int	Offset1	Offset type specification	
		Date format	0: 「Day」
			1: 「Month」
			2: 「Year」
		Time format	Time offset

int	Offset2	Specifying the Offset Range	
		Date format	Day Offset: 1 to 30
			Month Offset: 1 to 11
			Year offset: 1 to 98
		Time format	Minute offset

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for date and time formats and offsets.

3.11. GetObjectList

Get the object list

Declaration:

bool   LayoutDesigner.LibApi.GetObjectList(ref string[] object_list, ref int object_count)

Argument:

string[]	object_list	Object list
int	object_count;	Number of objects

Return value:

true	Normal end
false	Error

3.12. DelObjectData

Delete object data.

Declaration:

bool   LayoutDesigner.LibApi.DelObjectData(string object)

Argument:

string	object	Name of the object

Return value:

true	Normal end
false	Error

3.13. GetTextData

Gets object data of text.

Declaration:

bool   LayoutDesigner.LibApi.GetTextData(string object, ref Rectangle rect, ref string TextData,
ref int TextSource, ref string FontName, ref int FontPoint, ref int FontBold,
ref int FontItalic, ref int FontUnderline, ref int FontStrikeout, ref int FontCharset,
ref int TextAlign, ref bool AutoSize, ref int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Obtaining object coordinates
string	TextData	Get text data
int	TextSource	Get Data Source
string	FontName	Get a Font typeface
int	FontPoint	(Font) Get Point Size
int	FontBold	(Font) Get Enhanced Modifications
int	FontItalic	(Font) Get Italic Modifications
int	FontUnderline	(Font) Get Underline Qualification
int	FontStrikeout	(Font) Get Undo Line Qualification
int	FontCharset	(Font) Get Character Set
int	TextAlign	Get text alignment
bool	AutoSize	Get text auto-sizing
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for the modification of the Fonts, text alignment, and rotation angle.

3.14. SetTextData

Set object data for text.

Declaration:

bool　　　　LayoutDesigner.LibApi.SetTextData(ref string object, Rectangle rect, string TextData,
　　　　　　　　　　int TextSource, string FontName, int FontPoint, int FontBold, int FontItalic,
　　　　　　　　　　int FontUnderline, int FontStrikeout, int FontCharset, int TextAlign, bool AutoSize,
　　　　　　　　　　int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Specifying object coordinates
string	TextData	Specification of text data
int	TextSource	Specifying a Data Source
		0: Text data
		1: Date/Time
		2: Serial number
string	FontName	Specify font typeface
int	FontPoint	(Font) Specify Point Size
int	FontBold	(Font)Specifying Enhancement Modifications
		0: Disable　　1: Enable
int	FontItalic	(Font) Specifying Italic Modifications
		0: Disable　　1: Enable
int	FontUnderline	(Font) Specifying Underline Modifications
		0: Disable　　1: Enable
int	FontStrikeout	(Font) Designation of undo line modification
		0: Disable　　1: Enable
int	FontCharset	(Font) Specifying a Character Set
		0: European sentence
		128: Japanese
int	TextAlign	Specifying text alignment
		0: Left justified
		1: Centering
		2: Right justified
bool	AutoSize	Specifying automatic text sizing
		true:　　Enable
		false:　　Disable
int	angle	Specifying the rotation angle
		0: Rotation angle 0 degrees
		1: Rotation angle 90 degrees
		2: Rotation angle 180 degrees
		3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for the modification of the Fonts, text alignment, and rotation angle.

3.15. GetPTextData

Gets the object data of the printer text.

Declaration:

bool   LayoutDesigner.LibApi.GetPTextData(string object, ref Rectangle rect, ref string TextData,
                    ref int TextSource, ref int PrinterFont, ref int FontBold, ref int FontItalic,
                    ref int FontUnderline, ref int WidthSize, ref int HeightSize, ref int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Obtaining object coordinates
string	TextData	Get text data
int	TextSource	Get Data Source
int	PrinterFont	Get Printer Text Font Type
int	FontBold	(Printer Text) Get Enhanced Modifications
int	FontItalic	(Printer text) Get Italic Modifications
int	FontUnderline	(Printer text) Get Underline Qualification
int	WidthSize	(Printer text) Get horizontal double-angle size
int	HeightSize	(Printer text) Get vertical double-angle size
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for modification and size of printer text and rotation angle.

3.16. SetPTextData
Set object data for printer text

Declaration:
bool   LayoutDesigner.LibApi.SetPTextData(ref string object, Rectangle rect, string TextData,
                int TextSource, int PrinterFont, int FontBold, int FontItalic, int FontUnderline,
                int WidthSize, int HeightSize, int angle)

Argument:
string	object	Name of the object
Rectangle	rect	Specifying object coordinates
string	TextData	Specification of text data
int	TextSource	Specifying a Data Source
		0: Text data
		1: Date/Time
		2: Serial number
int	PrinterFont	Specifying the Printer Text Font Type
		0: Font A (24 dots)
		1: Font B (16 dots)
int	FontBold	(Printer Text) Specifying Enhancement Modifications
		0: Disable 1: Enable
int	FontItalic	(Printer text) specifying italic modification
		0: Disable 1: Enable
int	FontUnderline	(Printer Text) Specifying Underline Qualification
		0: Disable 1: Enable
int	WidthSize	(Printer text) Specifying the horizontal double-angle size
int	HeightSize	(Printer text) Designation of vertical double-angle size
		1 to 8: 1 to 8 times
int	angle	Specifying the rotation angle
		0: Rotation angle 0 degrees
		1: Rotation angle 90 degrees
		2: Rotation angle 180 degrees
		3: Rotation angle 270 degrees

Return value:
true	Normal end
false	Error

Note::
See Constant definitions in Chapter 4 for modification and size of printer text and rotation angle.

3.17. GetImageData

The object data of the image is acquired.

Declaration:

bool   LayoutDesigner.LibApi.GetImageData(string object, ref Rectangle rect, ref Bitmap bmp,

                    ref bool aspect, ref int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Obtaining object coordinates
Bitmap	bmp	Acquisition of Bitmap Data
bool	aspect	Get Aspect Ratio Settings (true or false)
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for aspect ratio and rotation angle.

3.18. SetImageData

Set the object data of the image.

Declaration:

bool   LayoutDesigner.LibApi.SetImageData(ref string object, Rectangle rect, string name, int type,
            int threshold, bool aspect, int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Specifying object coordinates
string	name	Name of the image file
int	type	Specification of conversion type of binarization
		0: Simplified binarization
		1: Dither transformation
		2: Error diffusion
int	threshold	Designation of binarization threshold (0 to 256)
bool	aspect	Specifying the Aspect Ratio Setting
		true: Maintain the aspect ratio
		false: Do not maintain aspect ratio
int	angle	Specifying the rotation angle
		0: Rotation angle 0 degrees
		1: Rotation angle 90 degrees
		2: Rotation angle 180 degrees
		3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for binarization, aspect ratio and rotation angle.

3.19. GetLineData

Gets the object data of a straight line.

Declaration:

bool   LayoutDesigner.LibApi.GetLineData(string object, ref Point st, ref Point ed, ref int width,

ref int type, ref int angle)

Argument:

string	object	Name of the object
Point	st	Get start coordinate data
Point	ed	Acquisition of End Coordinate Data
int	width	Obtaining the line width
int	type	Obtaining the line type
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for line width, line type, and rotation angle.

3.20. SetLineData

Sets the object data of a straight line.

Declaration:

bool   LayoutDesigner.LibApi.SetLineData(ref string object, Point st, Point ed, int width, int type, int angle)

Argument:

string	object	Name of the object
Point	st	Designation of start coordinate data
Point	ed	Designation of end coordinate data
int	width	Designation of line width (1 to 10 dots)
int	type	Designation of line type

                              0: Solid line

                              1: Dashed line

                              2: Dotted line

                              3: One-dot chain line

                              4: Two-dot chain line

int	angle	Specifying the rotation angle

                              0: Rotation angle 0 degrees

                              1: Rotation angle 90 degrees

                              2: Rotation angle 180 degrees

                              3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for line width, line type, and rotation angle.

3.21. GetRectData

Gets rectangular object data.

Declaration:

bool   LayoutDesigner.LibApi.GetRectData(string object, ref Rectangle rect, ref int width,
                                        ref int type, ref int angle)

Argument:

string   object   Name of the object
Rectangle   rect   Obtaining object coordinates

int   width   Obtaining the line width
int   type   Obtaining the line type
int   angle   Get rotation angle

Return value:

true   Normal end
false   Error

Note:

See Constant definitions in Chapter 4 for line width, line type, and rotation angle.

3.22. SetRectData

Set rectangular object data.

Declaration:

bool   LayoutDesigner.LibApi.SetRectData(ref string object, Rectangle rect, int width, int type, int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Specifying object coordinates
int	width	Designation of line width (1 to 10 dots)
int	type	Designation of line type

    0: Solid line

    1: Dashed line

    2: Dotted line

    3: One-dot chain line

    4: Two-dot chain line

int	angle	Specifying the rotation angle

    0: Rotation angle 0 degrees

    1: Rotation angle 90 degrees

    2: Rotation angle 180 degrees

    3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for line width, line type, and rotation angle.

3.23. GetFillData

Gets fill object data.

Declaration:

bool　　　　LayoutDesigner.LibApi.GetFillData(string object, ref Rectangle rect, ref int color, ref int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Obtaining object coordinates
int	color	Get Fill Color
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for fill color and rotation angle.

3.24. SetFillData

Set the fill object data.

Declaration:

bool　　　　LayoutDesigner.LibApi.SetFillData(ref string object, Rectangle rect, int color, int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Specifying object coordinates
int	color	Specify fill color
		0: Fill color Black
		1: Fill color White
		2: Fill color Invert
int	angle	Specifying the rotation angle
		0: Rotation angle 0 degrees
		1: Rotation angle 90 degrees
		2: Rotation angle 180 degrees
		3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for fill color and rotation angle.

3.25. GetBarcode1dData

Obtains the object data of the barcode.

Declaration:

bool   LayoutDesigner.LibApi.GetBarcode1dData(string object, ref Rectangle rect, ref string TextData,
                    ref int BarcodeSource, ref int BarcodeType, ref int HriPos, ref int BarcodeWidth,
                    ref int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Obtaining object coordinates
string	TextData	Get text data
int	BarcodeSource	Obtaining a data source for a barcode
int	BarcodeType	Obtaining barcode type
int	HriPos	Obtaining the printing position of HRI characters
int	BarcodeWidth	Get Bar Width
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for barcode data sources, types, widths, HRI characters and rotation angles.

3.26. SetBarcode1dData

Sets the object data of the barcode.

Declaration:

bool   LayoutDesigner.LibApi.SetBarcode1dData(ref string object, Rectangle rect, string TextData,
int BarcodeSource, int BarcodeType, int HriPos, int BarcodeWidth, int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Specifying object coordinates
string	TextData	Specification of text data
int	BarcodeSource	Specifying the barcode data source
		0: Text data
		1: Serial number
int	BarcodeType	To specify the bar code type.
		1: UPC-A
		2: UPC-E
		3: JAN13
		4: JAN8
		5: CODE39
		6: ITF
		7: CODABAR
		8: CODE128
		9: CODE93
int	HriPos	Designation of print position of HRI character
		0: Do not use HRI characters
		1: Align HRI Characters on the Upper Side of the Bar Code
		2: Align HRI Characters to the Bottom of the Bar Code
		3: Align HRI characters above and below the bar code
int	BarcodeWidth	Specifying the bar width
		0 to 3: 1 to 4 dots
int	angle	Specifying the rotation angle
		0: Rotation angle 0 degrees
		1: Rotation angle 90 degrees
		2: Rotation angle 180 degrees
		3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for barcode data sources, types, widths, HRI characters and rotation angles.

3.27. GetBarcode2dData

Obtains the object data of the 2D code.

Declaration:

bool   LayoutDesigner.LibApi.GetBarcode2dData(string object, ref Rectangle rect, ref string TextData,
            ref int BarcodeSource, ref int BarcodeType, ref int CellSize, ref int EccLevel, ref int SymbolSize,
            ref int Colum, ref int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Obtaining object coordinates
string	TextData	Get text data
int	BarcodeSource	Get 2D Code Data Sources
int	BarcodeType	Acquisition of 2D Code Type
int	CellSize	Get cell size for QR code
int	EccLevel	Obtaining ECC levels
int	SymbolSize	Get symbol size
int	Colum	Get column count (discard except Micro PDF417 selection)
int	angle	Get rotation angle

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for the data source, type, cell size, ECC level, symbol size, column number and rotation angle of the 2D code.

3.28. SetBarcode2dData

Set the object data of the 2D code.

Declaration:

bool   LayoutDesigner.LibApi.SetBarcode2dData(ref string object, Rectangle rect, string TextData,
                 int BarcodeSource, int BarcodeType, int CellSize, int EccLevel, int SymbolSize,
                 int Colum, int angle)

Argument:

string	object	Name of the object
Rectangle	rect	Specifying object coordinates
string	TextData	Specification of text data
int	BarcodeSource	Specifying the 2D Code Data Source
		0: Text data
		1: Serial number
int	BarcodeType	2D Code Type Specification
		1: QR CODE
		2: Micro QR CODE
		3: PDF417
		4: Micro PDF417
		5: DATA MATRIX
		6: MAXI CODE
int	CellSize	To specify the cell size of the QR code.
		1 to 8: Number of dots per 1 cell
int	EccLevel	Specifying the ECC Level
		1: ECC level 7%　(QR CODE /Micro QR CODE)
		2: ECC level 15%　(QR CODE /Micro QR CODE)
		3: ECC level 25%　(QR CODE /Micro QR CODE)
		4: ECC level 30%　(QR CODE)
int	SymbolSize	Designation of symbol size
		When QR CODE is selected
		0: Auto size　　1 to 40: Specified size
		When Micro QR CODE is selected
		0: Auto size　　1 to 4: Specified size
		When DATA MATRIX is selected
		Specified size of 10, 18, 22, 26, 32, 40, 48
int	Colum	Number of columns specified
		(Value enabled only when Micro PDF417 is selected)
		0 to 3: Number of columns 1 to 4 is specified.
int	angle	Specifying the rotation angle
		0: Rotation angle 0 degrees
		1: Rotation angle 90 degrees
		2: Rotation angle 180 degrees
		3: Rotation angle 270 degrees

Return value:

true	Normal end
false	Error

Note:

See Constant definitions in Chapter 4 for the data source, type, cell size, ECC level, symbol size, column number and rotation angle of the 2D code.

3.29. GetMacroData

Gets the object data of the document macro.

Declaration:

bool   LayoutDesigner.LibApi.GetMacroData(string object, ref string[] start_cmd, ref string[] end_cmd)

Argument:

string	object	Name of the object
string[]	start_cmd	Get Page Start Command
string[]	end_cmd	Get Page End Command

Return value:

true	Normal end
false	Error

3.30. SetMacroData

Set the object data of the document macro.

Declaration:

bool   LayoutDesigner.LibApi.SetMacroData(ref string object, string[] start_cmd, string[] end_cmd)

Argument:

string	object	Name of the object
string[]	start_cmd	Page Start Command Specification
string[]	end_cmd	Designation of Page End Command

Return value:

true	Normal end
false	Error

4. Constant definition

The list of error constant definitions for the LibApi class is as follows:

Name	Value	Item
Error type		
ERROR_CODE_OK	0	Normal end
ERROR_CODE_VER_GET	1	Failed to acquire version
ERROR_CODE_SDK_INIT	2	SDK initialization failed
ERROR_CODE_FILE_NEW	3	Failed to create new SOP file
ERROR_CODE_PAPER_SETTING	4	Paper setting failed
ERROR_CODE_FILE_READ	5	Failed to load SOP file
ERROR_CODE_FILE_WRITE	6	Failed to write SOP file
ERROR_CODE_FILE_PRINT	7	Failed to print SOP file
ERROR_CODE_FILE_PREVIEW	8	SOP file preview creation failed
ERROR_CODE_OBJECT_LIST_GET	9	Failed to acquire object list
ERROR_CODE_OBJECT_DATA_GET	10	Failed to acquire object data
ERROR_CODE_OBJECT_DATA_SET	11	Setting of object data failed
ERROR_CODE_OBJECT_DATA_DEL	12	Object data deletion failed
ERROR_CODE_SERIAL_INIT	13	Serial number initialization failed
ERROR_CODE_SERIAL_SET	14	Serial number setting failed
ERROR_CODE_DATE_TIME_SET	15	Failed to set date/time data
ERROR_CODE_TEXT_OBJECT_GET	16	Text data acquisition failed
ERROR_CODE_TEXT_OBJECT_SET	17	Setting of text data failed
ERROR_CODE_PTEXT_OBJECT_GET	18	Failed to acquire printer text data
ERROR_CODE_PTEXT_OBJECT_SET	19	Setting of printer text data failed
ERROR_CODE_IMAGE_OBJECT_GET	20	Failed to acquire image data
ERROR_CODE_IMAGE_OBJECT_SET	21	Failed to set image data
ERROR_CODE_LINE_OBJECT_GET	22	Failed to acquire linear data
ERROR_CODE_LINE_OBJECT_SET	23	Setting of linear data failed.
ERROR_CODE_RECT_OBJECT_GET	24	Failed to acquire rectangular data
ERROR_CODE_RECT_OBJECT_SET	25	Setting of rectangular data failed
ERROR_CODE_FILL_OBJECT_GET	26	Failed to acquire fill data
ERROR_CODE_FILL_OBJECT_SET	27	Failed to set fill data
ERROR_CODE_BARCODE_1D_OBJECT_GET	28	Failed to acquire one-dimensional barcode data
ERROR_CODE_BARCODE_1D_OBJECT_SET	29	Setting of one-dimensional barcode data failed
ERROR_CODE_BARCODE_2D_OBJECT_GET	30	Failed to acquire 2-D barcode data
ERROR_CODE_BARCODE_2D_OBJECT_SET	31	Setting of 2-D barcode data failed
ERROR_CODE_MACRO_OBJECT_GET	32	Failed to acquire document macro data
ERROR_CODE_MACRO_OBJECT_SET	33	Document macro data setting failed

The list of constant definitions that can be specified in the API of the LibApi class is as follows.

Name	Value	Item
Printer model number		
PRINTER_SK1_21	1	SK1-21、SK1-211
PRINTER_SK1_31	2	SK1-31、SK1-311
PRINTER_SK1_41	3	SK1-41
PRINTER_SK4_21	4	SK4-21
PRINTER_SK4_31	5	SK4-31
PRINTER_BL2_58	6	BL2-58
PRINTER_SD3_21	7	SD3-21
PRINTER_SD3_22	8	SD3-22
PRINTER_SM1_21	9	SM1-21
PRINTER_BLM_80	10	BLM-80
PRINTER_SM2_41	11	SM2-41
PRINTER_SM3_21	12	SM3-21
PRINTER_SM4_21	13	SM4-21
PRINTER_SM4_31	14	SM4-31
Object type		
OBJECT_TYPE_NONE	0	Invalid object
OBJECT_TYPE_TEXT	1	Text object
OBJECT_TYPE_PTEXT	2	Printer text object
OBJECT_TYPE_RECT	3	Rectangular object
OBJECT_TYPE_FILL	4	Fill object
OBJECT_TYPE_LINE	5	Linear object
OBJECT_TYPE_IMAGE	6	Image object
OBJECT_TYPE_BAR1	7	One-Dimensional barcode objects
OBJECT_TYPE_BAR2	8	Two-dimensional barcode object
OBJECT_TYPE_MACRO	9	Document macro object
Rotation angle		
OBJECT_ANGLE0	0	Rotation angle 0 degrees
OBJECT_ANGLE90	1	Rotation angle 90 degrees
OBJECT_ANGLE180	2	Rotation angle 180 degrees
OBJECT_ANGLE270	3	Rotation angle 270 degrees
Data source for text or printer text		
TEXT_SOURCE_TEXT	0	Text data
TEXT_SOURCE_DATE	1	Date/Time
TEXT_SOURCE_SERIAL	2	Serial number
Enhancement modification of text or printer text		
TEXT_BOLD_ENABLE	0	Enable the enhancement modification
TEXT_BOLD_DISABLE	1	Disable the enhancement modification
Italic modification of text or printer text		
TEXT_ITALIC_ENABLE	0	Enable italic modification
TEXT_ITALIC_DISABLE	1	Disable italic modification
Underline modification of text or printer text		
TEXT_UNDERLINE_ENABLE	0	Enable underline modification
TEXT_UNDERLINE_DISABLE	1	Disable the underline modification
Text Font Undo Line Qualification		
TEXT_STRIKE_ENABLE	0	Enable undo line modification
TEXT_STRIKE_DISABLE	1	Disable undo line modification
Character set of a Text character		
CHARACTER_SET_JAPAN	128	Character set Japanese
CHARACTER_SET_EURO	0	Character set European sentence

Text Font Alignment		
TEXT_ALIGN_LEFT	0	Left justified
TEXT_ALIGN_CENTER	1	centering
TEXT_ALIGN_RIGHT	2	Right justified
Auto-Size Text Fonts		
TEXT_AUTO_SIZE_DISABLE	false	Disable automatic sizing
TEXT_AUTO_SIZE_ENABLE	true	Enable Auto Size Adjustment
Printer Text Font Type		
PTEXT_FONTA	0	Font A (24 dots)
PTEXT_FONTB	1	Font B (16 dots)
Double-angle size of the printer text		
PTEXT_ZOOM_SIZE1	1	1x
PTEXT_ZOOM_SIZE2	2	2x
PTEXT_ZOOM_SIZE3	3	3x
PTEXT_ZOOM_SIZE4	4	4x
PTEXT_ZOOM_SIZE5	5	5x
PTEXT_ZOOM_SIZE6	6	6x
PTEXT_ZOOM_SIZE7	7	7x
PTEXT_ZOOM_SIZE8	8	8x
Binary conversion type		
IMAGE_CONVERT_SIMPLE	0	Simplified binarization
IMAGE_CONVERT_DITHER	1	Dither transformation
IMAGE_CONVERT_ERROR_DIFFUSION	2	Error diffusion
Binarization threshold		
IMAGE_THRESHOLD_MIN	0	Threshold minimum
IMAGE_THRESHOLD_MAX	256	Threshold maximum
Aspect Ratio		
ASPECT_RATIO_NON_KEEP	false	Do not maintain aspect ratio
ASPECT_RATIO_KEEP	true	Maintain the aspect ratio
Line type		
LINE_TYPE_SOLID	0	Line type Solid line
LINE_TYPE_DASH	1	Type Dashed line
LINE_TYPE_DOT	2	Line type Dotted line
LINE_TYPE_1DOT_CHAIN	3	Line type 1-dot chain line
LINE_TYPE_2DOT_CHAIN	4	Line type 2-dot chain line
Line width		
LINE_WIDTH_1DOT	1	Line width 1 dot
LINE_WIDTH_2DOT	2	Line width 2 dots
LINE_WIDTH_3DOT	3	Line width 3 dots
LINE_WIDTH_4DOT	4	Line width 4 dots
LINE_WIDTH_5DOT	5	Line width 5 dots
LINE_WIDTH_6DOT	6	Line width 6 dots
LINE_WIDTH_7DOT	7	Line width 7 dots
LINE_WIDTH_8DOT	8	Line width 8 dots
LINE_WIDTH_9DOT	9	Line width 9 dots
LINE_WIDTH_10DOT	10	Line width 10 dots
Fill color		
FILL_COLOR_BLACK	0	Fill color Black
FILL_COLOR_WHITE	1	Fill color White
FILL_COLOR_REVERSE	2	Fill color Invert

Barcode type		
BARCODE1D_TYPE_UPCA	1	UPC-A
BARCODE1D_TYPE_UPCE	2	UPC-E
BARCODE1D_TYPE_JAN13	3	JAN13
BARCODE1D_TYPE_JAN8	4	JAN8
BARCODE1D_TYPE_CODE39	5	CODE39
BARCODE1D_TYPE_ITF	6	ITF
BARCODE1D_TYPE_CODABAR	7	CODABAR
BARCODE1D_TYPE_CODE128	8	CODE128
BARCODE1D_TYPE_CODE93	9	CODE93
2D Code Types		
BARCODE2D_TYPE_QRCODE	1	QR CODE
BARCODE2D_TYPE_MICRO_QRCODE	2	Micro QR CODE
BARCODE2D_TYPE_PDF417	3	PDF417
BARCODE2D_TYPE_MICRO_PDF417	4	Micro PDF417
BARCODE2D_TYPE_DATA_MATRIX	5	DATA MATRIX
BARCODE2D_TYPE_MAXI_CODE	6	MAXI CODE
Barcode and 2D code data sources		
BARCODE_SOURCE_TEXT	0	Text data
BARCODE_SOURCE_SERIAL	1	Serial number
Printing position of HRI letters		
HRI_LOCATE_NONE	0	Do not use HRI characters
HRI_LOCATE_UPPER	1	Align HRI Characters on the Upper Side of the Barcode
HRI_LOCATE_BOTTOM	2	Align HRI Characters to the Bottom of the Barcode
HRI_LOCATE_BOTH	3	Align HRI characters above and below the Barcode
Bar width		
BAR_WIDTH1	0	Bar width 1 dot
BAR_WIDTH2	1	Bar width 2 dots
BAR_WIDTH3	2	Bar width 3 dots
BAR_WIDTH4	3	Bar width 4 dots
Printing position of HRI letters		
HRI_LOCATE_NONE	0	Do not use HRI characters
HRI_LOCATE_UPPER	1	Align HRI Characters on the Upper Side of the Barcode
HRI_LOCATE_BOTTOM	2	Align HRI Characters to the Bottom of the Barcode
HRI_LOCATE_BOTH	3	Align HRI characters above and below the Barcode
Cell size of the QR code		
QRCODE_CELL_SIZE1	1	Cell size 1 of QR CODE
QRCODE_CELL_SIZE2	2	Cell size 2 of QR CODE
QRCODE_CELL_SIZE3	3	Cell size 3 of QR CODE
QRCODE_CELL_SIZE4	4	Cell size 4 of QR CODE
QRCODE_CELL_SIZE5	5	Cell size 5 of QR CODE
QRCODE_CELL_SIZE6	6	Cell size 6 of QR CODE
QRCODE_CELL_SIZE7	7	Cell size 7 of QR CODE
QRCODE_CELL_SIZE8	8	Cell size 8 of QR CODE

Symbol size			
QRCODE_SYMBOL_MIN_SIZE	0	QR CODE auto size	
QRCODE_SYMBOL_MAX_SIZE	40	QR CODE maximum size	
MICRO_QRCODE_SYMBOL_MIN_SIZE	0	MICRO QR CODE auto size	
MICRO_QRCODE_SYMBOL_MAX_SIZE	4	MICRO QR CODE maximum size	
DATAMATRIX_SYMBOL_SIZE10	10	DATA MATRIX size 10	
DATAMATRIX_SYMBOL_SIZE18	18	DATA MATRIX size 18	
DATAMATRIX_SYMBOL_SIZE22	22	DATA MATRIX size 22	
DATAMATRIX_SYMBOL_SIZE26	26	DATA MATRIX size 26	
DATAMATRIX_SYMBOL_SIZE32	32	DATA MATRIX size 32	
DATAMATRIX_SYMBOL_SIZE40	40	DATA MATRIX size 40	
DATAMATRIX_SYMBOL_SIZE48	48	DATA MATRIX size 48	
ECC level			
QRCODE_ECC_L	1	QR CODE ECC Level 7%	
QRCODE_ECC_M	2	QR CODE ECC level 15%	
QRCODE_ECC_Q	3	QR CODE ECC level 25%	
QRCODE_ECC_H	4	QR CODE ECC level 30%	
Number of rows			
MICRO_PDF417_COLUMNS_SIZE1	0	Column number 1 of Micro PDF417	
MICRO_PDF417_COLUMNS_SIZE2	1	Column number 2 of Micro PDF417	
MICRO_PDF417_COLUMNS_SIZE3	2	Column number 3 of Micro PDF417	
MICRO_PDF417_COLUMNS_SIZE4	3	Column number 4 of Micro PDF417	
Object type with serial number			
OBJECT_TYPE_TEX	0	Text	
OBJECT_TYPE_BARCODE	1	Barcode	
Serial number type			
SERIAL_TYPE_DECIMAL	0	Decimal number	
Increasing/decreasing serial number			
SERIAL_NUMBER_INC	0	Sequential number increase	
SERIAL_NUMBER_DEC	1	Sequential decrease	
Selection of date or time			
SELECT_TYPE_DATE	0	Date selection	
SELECT_TYPE_TIME	1	Time selection	
Date offset type			
OFFSET_TYPE_DAY	0	Day	
OFFSET_TYPE_MONTH	1	Month	
OFFSET_TYPE_YEAR	2	Year	
Date offset range			
OFFSET_DAY_MIN	1	Daily offset minimum	
OFFSET_DAY_MAX	30	Maximum daily offset	
OFFSET_MONTH_MIN	1	Minimum offset value for the month	
OFFSET_MONTH_MAX	11	Maximum offset value for the month	
OFFSET_YEAR_MIN	1	Year's minimum offset	
OFFSET_YEAR_MAX	98	Year's maximum offset	

Date format		
DATE_FORMAT_TYPE01	0	2021/04/13(Year/Month/Day)
DATE_FORMAT_TYPE02	1	21/04/13(Year/Month/Day)
DATE_FORMAT_TYPE03	2	2021-04-13(Year-Month-Day)
DATE_FORMAT_TYPE04	3	21-04-13(Year-Month-Day)
DATE_FORMAT_TYPE05	4	2021,04,13(Year,Month,Day)
DATE_FORMAT_TYPE06	5	21,04,13(Year,Month,Day)
DATE_FORMAT_TYPE07	6	2021.04.13(Year.Month.Day)
DATE_FORMAT_TYPE08	7	21.04.13(Year,Month.Day)
DATE_FORMAT_TYPE09	8	April 13, 2021
DATE_FORMAT_TYPE10	9	April 13, 210
DATE_FORMAT_TYPE11	10	April 13, 2021 Tuesday
DATE_FORMAT_TYPE12	11	April 13, 210 Tuesday
DATE_FORMAT_TYPE13	12	April 13, 2003
DATE_FORMAT_TYPE14	13	April 13, R03
DATE_FORMAT_TYPE15	14	R03/04/13(Year/Month/Day)
DATE_FORMAT_TYPE16	15	R03-04-13(Year-Month-Day)
DATE_FORMAT_TYPE17	16	R03,04,13(Year,Month,Day)
DATE_FORMAT_TYPE18	17	R03.04.13(Year.Month.Day)
DATE_FORMAT_TYPE19	18	210413(YearMonthDay)
DATE_FORMAT_TYPE20	19	20210413(YearMonthDay)
DATE_FORMAT_TYPE21	20	Fire (DayOfWeek)
DATE_FORMAT_TYPE22	21	21(Year)
DATE_FORMAT_TYPE23	22	2021(Year)
DATE_FORMAT_TYPE24	23	04(Month)
DATE_FORMAT_TYPE25	24	13(Day)
Time format		
TIME_FORMAT_TYPE01	0	02:22 PM(12 Hour Format)
TIME _FORMAT_TYPE02	1	02:22(12 Hour Format)
TIME _FORMAT_TYPE03	2	02:22:19(12 Hour Format)
TIME _FORMAT_TYPE04	3	02:22:19(12 Hour Format:2Digits)
TIME _FORMAT_TYPE05	4	02:22:19 PM(12 Hour Format)
TIME _FORMAT_TYPE06	5	14:22 PM(24 Hour Format)
TIME _FORMAT_TYPE07	6	14:22(24 Hour Format)
TIME _FORMAT_TYPE08	7	14:22:19(24 Hour Format)
TIME _FORMAT_TYPE09	8	14:22:19(24 Hour Format:2Digits)
TIME _FORMAT_TYPE10	9	02:22 (12 Hour Format)
TIME _FORMAT_TYPE11	10	14:22 (24 Hour Format)
TIME _FORMAT_TYPE12	11	02:22:19 (12 Hour Format)
TIME _FORMAT_TYPE13	12	14:22:19 (24 Hour Format)
TIME _FORMAT_TYPE14	13	02:22 p.m. (12 Hour Format)
TIME _FORMAT_TYPE15	14	14:22 p.m. (24 Hour Format)
TIME _FORMAT_TYPE16	15	PM
TIME _FORMAT_TYPE17	16	02:00 (12 Hour Format)
TIME _FORMAT_TYPE18	17	14:00 (24 Hour Format)
TIME _FORMAT_TYPE19	18	02:00(12 Hour Format)
TIME _FORMAT_TYPE20	19	14:00(24 Hour Format)

APPENDIX: SAMPLE PROGRAMS

The SDK comes with a sample program that creates the label image shown in the figure below.

Includes how to use each API.

See Chapter 2 for sample configuration and processes.