SII

DPU-S445 SERIES
Microsoft® .NET™ Compact Framework
PRINT CLASS LIBRARY
TECHNICAL REFERENCE

Seiko Instruments Inc.

DPU-S445 SERIES Microsoft® .NET™ Compact Framework PRINT CLASS LIBRARY TECHNICAL REFERENCE

First Edition September 2008

PREFACE

This document describes the specifications and functions of Microsoft® .NET Compact Framework Print Class Libraries, SII DPU-S445 NETCF10 and SII DPU-S445 NETCF20 for the thermal printer DPU-S445 series offered by Seiko Instruments Inc. (SII).

TABLE OF CONTENTS

Section Page

CHAPTER 5
Sample Program

CHAPTER 6
Disclaimer

CHAPTER 1
PRODUCT OVERVIEW

This chapter provides the overview of SII DPU-S445 NETCF10 and SII DPU-S445 NETCF20 (These libraries).

1.1 FUNCTION PROVIDED BY THESE LIBRARIES

These libraries offer functions for applications working in the Microsoft® .NET Compact Framework (.NET Compact Framework) environment to use the DPU-S445 series from SII (Printer). (Figure 1-1)

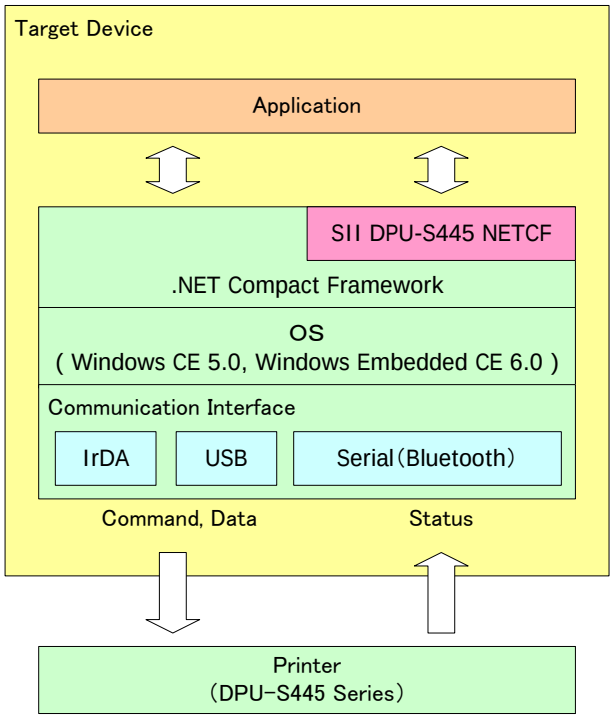


Figure 1-1

Applications can send commands and data to a printer through the communication interface (IrDA, USB, and serial) to the target device by these libraries. In addition, the printer status can be obtained.

These libraries offer the following functions:

- Connection/disconnection to/from a printer
- Sending data to a printer (printing data or commands*)
- Sending a data file to a printer (printing data or commands*)
- Obtaining the printer status
- Aborting the data-wait status of the printer

*: It does not support the command to acquire the response from the printer.

1.2 SUPPORTED .NET COMPACT FRAMEWORK

1.2.1 .NET Compact Framework version and supporting product

These libraries depend on supported versions of the .NET Framework (.NET Compact Framework 1.0 and .NET Compact Framework 2.0). Table 1-1 lists the versions of the .NET Compact Framework and supporting products.

Table 1-1

Product	Supported .NET Compact Framework
SII DPU-S445 NETCF10	.NET Compact Framework 1.0
SII DPU-S445 NETCF20	.NET Compact Framework 2.0

SII DPU-S445 NETCF10 supports .NET Compact Framework 1.0, not .NET Compact Framework 2.0. Similarly, SII DPU-S445 NETCF20 supports .NET Compact Framework 2.0, not .NET Compact Framework 1.0. (Figure 1-2)

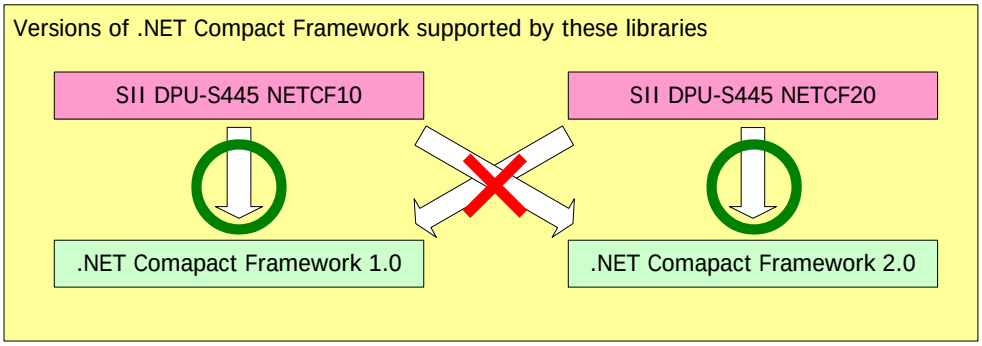


Figure 1-2

In addition, this manual describes these libraries as SII DPU-S445 NETCF if the item is common to each version for .NET Compact Framework.

CHAPTER 2
PRODUCT SPECIFICATIONS

This chapter describes the product specifications for these libraries.

2.1 PRODUCT SPECIFICATIONS

2.1.1 Operational standard

Table 2-1 shows the operational standard for these libraries.

Table 2-1

Items	Specifications
Target OS	Windows CE 5.0 Windows Embedded CE 6.0 (including Windows Mobile 5.0 and Windows Mobile 6)
Target .NET Compact Framework	.NET Compact Framework 1.0 (SP3 or later) .NET Compact Framework 2.0 (SP1 or later)
Supported languages	Japanese English
Target communication method	Infrared (IrDA DATA 1.2) Serial (Virtual serial port using RS-232C and Bluetooth (SPP)) USB 2.0
Supported development languages	Languages available for developing the .NET Compact Framework Visual C# .NET 2003 (only DPU-S445 NETCF10) Visual C# 2005(DPU-S445 NETCF10 and DPU-S445 NETCF20) Visual Basic .NET 2003(Only DPU-S445 NETCF10) Visual Basic 2005(DPU-S445 NETCF10 and DPU-S445 NETCF20)

2.1.2 Details on communication methods

(1) IrDA DATA communication

The maximum baud rate is 115200 bps (SIR). Application protocol is IrLPT.

(2) Serial communication

Table 2-2 shows the specifications for serial communication.

Table 2-2

Items	Specifications
Baud rate	9600, 38400, 57600 and115200 bps
Byte size	8 bits (fixed)
Stop bit	1 bit (fixed)
Parity	None (fixed)
Flow control	Xon/Xoff, hardware (CTS/RTS)

(3) Bluetooth communication

Bluetooth communication is available when a connection is established as a virtual serial port with SPP (Serial Port Profile).

(4) USB communication

USB communication requires that target device should be support to the USB host driver and USB Printer class driver (isbprn.dll) is implement on a target OS.

USB Printer class driver is available for the LPT port on the target OS for these libraries.

USB Printer class driver does not implemented to these libraries. USB Printer class driver is available from the manufacturers which have provided the target device using these libraries, or build USB Printer class driver for the target device using Platform Builder Microsoft. USB Printer class driver should be used under the environment which has complied with the target device using these libraries.

2.2 PROVIDED FILES AND HOW TO USE THESE FILES

2.2.1 Provided files

These libraries are composed of the DLL file for the print class library and for serial communication.

Products for .NET Compact Framework 1.0 and .NET Compact Framework 2.0 are provided depending on the supported versions of the .NET Compact Framework.

Table 2-3 shows each product supporting each .NET Compact Framework version and the file configuration.

Table 2-3

Product	Supported .NET Compact Framework	File	Function
SII DPU-S445 NETCF10	.NET Compact Framework 1.0	**SiiDpuS445NetCF10.dll**	Class library
		SiiSerialNetCF10.dll	For serial communication
SII DPU-S445 NETCF20	.NET Compact Framework 2.0	**SiiDpuS445NetCF20.dll**	Class library
		SiiSerialNetCF20.dll	For serial communication

2.2.2 Embedding these libraries into a project

To use these libraries from Visual C# and Visual Basic, create a project and add either one of these libraries (**SiiDpuS445NetCF10.dll** or **SiiDpuS445NetCF20.dll**) to the reference setting for [Add reference]. (For Visual C#.NET 2003 and Visual Basic.NET 2003, only **SiiDpuS445NetCF10.dll** is available.)

2.2.3 How to execute a created application on a target device

Store an application created in Visual C# or Visual Basic in the same folder as that for these libraries.

To use .NET Compact Framework 1.0, store **SiiDpuS445NetCF10.dll** and **SiiSerialNetCF10.dll** in the same folder as the created application.

To use .NET Compact Framework 2.0, store **SiiDpuS445NetCF20.dll** and **SiiSerialNetCF20.dll** in the same folder as the created application.

2.2.4 Other

The following SDKs may be necessary depending on the program to be developed.

- Windows Mobile 5 Pocket PC SDK
- Windows Mobile 6 Standard SDK
- Windows Mobile 6 Professional SDK

These SDKs are available from the Microsoft website.

CHAPTER 3
INSTALLATION

This chapter describes the installation of these libraries.

3.1 INSTALLER

3.1.1 Installer

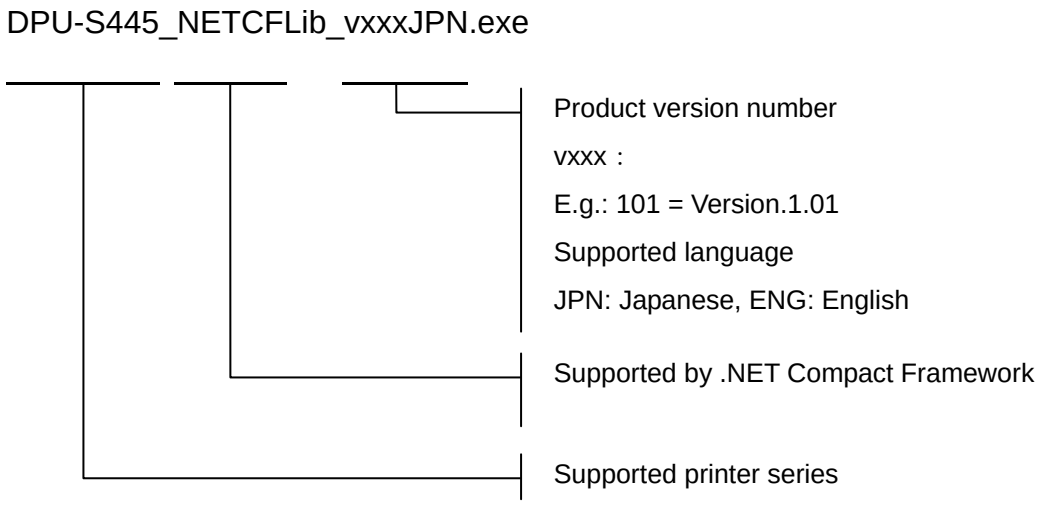
Two versions of the installer are provided for these class libraries: the Japanese and English versions (Table 3-1). The only difference between the versions is the language displayed in the dialog window when installing the file. The library files installed on the PC are the same.

Table 3-1

Supported language	File name
Japanese	DPU-S445_NETCFLib_vxxxJPN.exe
English	DPU-S445_NETCFLib_vxxxENG.exe

These libraries (DPU-S445 NETCF10 and DPU-S445 NETCF20) are installed on the PC when executing the installer.

The different parts of the installer file name have different meanings as shown below;



3.1.2 Installation file

Figure 3-1 shows the files that will be installed on the PC and the directory configuration after executing the installer.

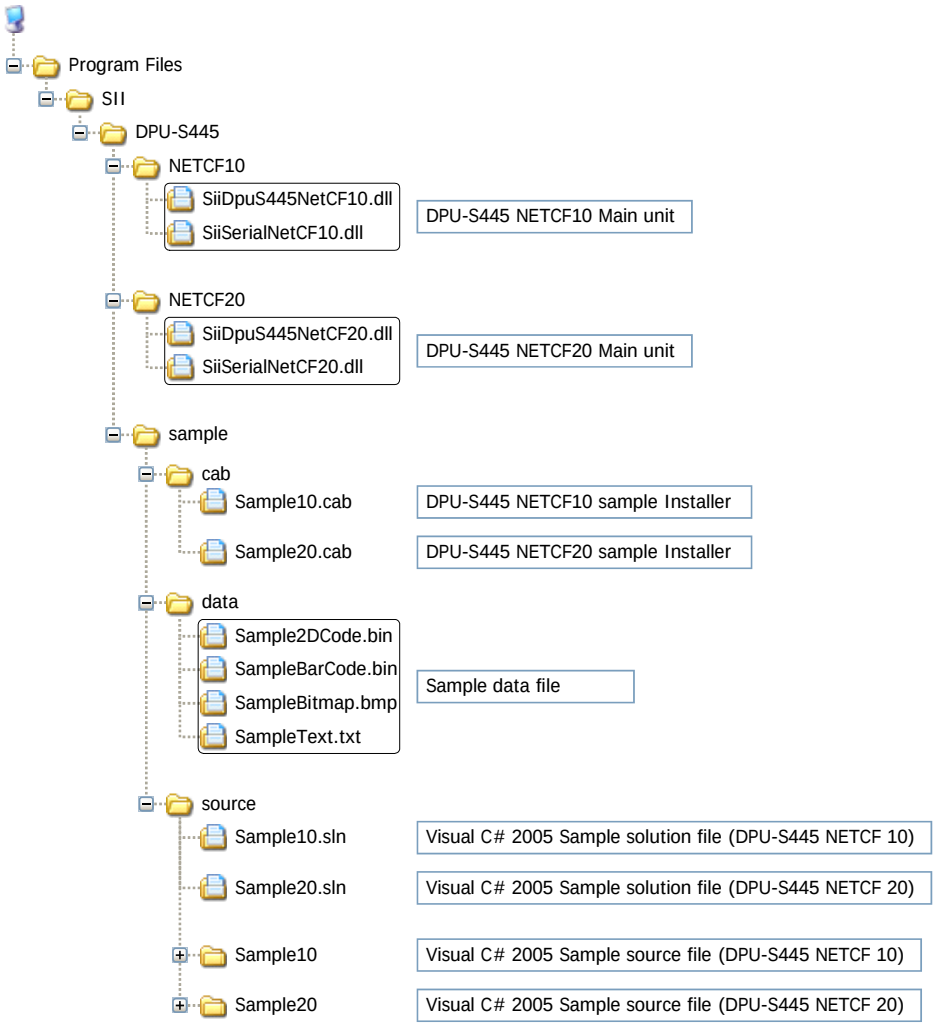


Figure 3-1

CHAPTER 4
PRINT CLASS LIBRARY FUNCTION

This chapter describes the class structure of this library and the methods and the properties implemented in the class.

The functions of this library are common between DPU-S445 NETCF10 and DPU-S445 NETCF20 unless otherwise specified.

For details on how to use this library, see the sample program (C#).

4.1 CLASS OVERVIEW

4.1.1 Name space and Class name

This library provides the following common name space and class name regardless of the supported .NET Compact Framework. Table 4-1 shows the name space and the class name.

Table 4-1

Item	Description
Name space	SII.SII_Print
Class name	DPU_S445
Fully Qualified Name	SII.SII_Printer.DPU_S445

4.1.2 Class structure

Figure 4-1 shows the structure of the DPU_S445 class (Hereinafter, This class).

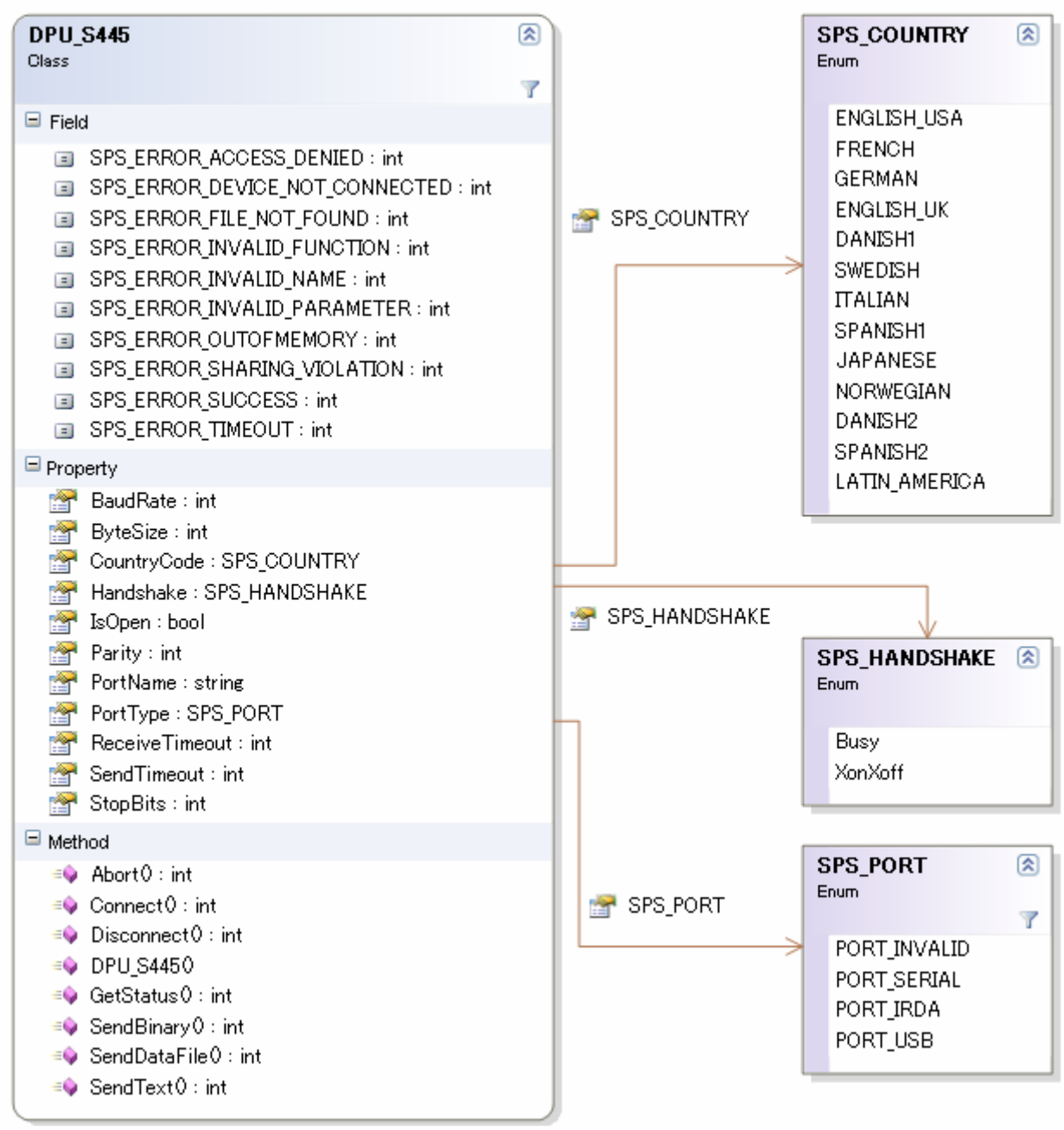


Figure 4-1

4.1.3 Method list

Table 4-2 shows the list of the methods implemented in this class.

Table 4-2

Method	Description	Page
DPU_S445	Constructor	4-4
Connect	Connection to a printer	4-5
Disconnect	Disconnection from a connected port	4-7
GetStatus	Obtains the status of a printer	4-8
SendText	Sends text data	4-10
SendBinary	Sends binary data	4-11
SendDataFile	Sends a data file	4-12
Abort	Aborts the status of waiting for data from a printer	4-15

4.1.4 Property list

Table 4-3 shows the list of properties implemented in this class.

Table 4-3

Property	Access	Description	Page
SendTimeout	R/W	Time-out period when data is sent	4-16
ReceiveTimeout	R/W	Time-out period when data is received	4-17
CountryCode	R/W	Country code	4-18
BaudRate	R/W	Baud rate during serial connection	4-20
Handshake	R/W	Flow control during serial connection	4-21
ByteSize	R	Byte size during serial connection	4-22
StopBits	R	Stop bit during serial connection	4-23
Parity	R	Parity during serial connection	4-24
PortName	R	Name of a connected port	4-25
PortType	R	Type of a connected port	4-26
IsOpen	R	Connection	4-27

4.2 METHOD DETAILS

This section provides details on the methods implemented in this class.

4.2.1 Constructor

DPU_S445()	Constructor

Creates a new object for this class.

Syntax

DPU_S445 ()

Parameter

None

Return value

None

Description

Creates a new object for this class.

Example: Declared as shown below to create an object in Visual C#.

```
using SII.SII_Print;
    •
    •
DPU_S445 prn = new DPU_S445();
```

Example: Declared as shown below to create an object in Visual Basic.

```
Imports SII.SII_Print
    •
    •
Dim prn As New DPU_S445
```

4.2.2 Connect method

Connect	Connection to a printer

Specifies a communication port used for this class and connects it to a printer.

Syntax

Int32 ***Connect*** (String *PortName*)

Parameter

PortName : Specifies the name of a port to be connected to a printer. It is String type.

Table 4-4 shows parameters effective as *PortName*

Table 4-4

PortName	**Port type**	**Description**
COMx:	Serial port	Connects to a printer through the serial port. Specify any among COM1:-COM9:. For virtual serial port connection using Bluetooth (SPP), specify COMx: with established communication using Bluetooth.
IR:	Infrared port (IrDA)	Connects to a printer using IrDA.
LPTx:	Printer port (USB)	Connects to a printer using the USB port. Specify any from LPT1: to LPT9:

Return value

Table 4-5 shows the values returned by this method.

Table 4-5

Constant	**Value**	**Description**
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_ACCESS_DENIED	5	Failed to obtain the handle.
		Unavailable port specified.
SPS_ERROR_SHARENG_VIOLATION	32	Already opened port specified.
		Another device opens the specified port.
SPS_ERROR_INVALID_NAME	123	No specified port exists.
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	No printer found (For IrDA).
SPS_ERROR_TIMEOUT	1467	Send or receive time-out.

Description

This method is called before the counterpart of another class is used.

This method opens a port specified with *PortName* and connects to a printer. For connection to ports other than the USB port, match the communication mode to a port type specified with *PortName* in the functional settings of the printer beforehand. For connection using the serial port, match the settings of the ***Baudrate*** property and the ***Handshake*** property with the communication settings of the printer.

This method is subject to change in order to use printer settings in these libraries during the connection.

See also

Baudrate property, ***Handshake*** property

4.2.3 Disconnect method

Disconnect	Disconnecting a connected port

Disconnects a connected port.

Syntax

Int32 ***Disconnect*** ()

Parameter

None

Return value

Table 4-6 shows the values returned by this method.

Table 4-6

Constant	Value	Description
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_INVALID_FUNCTION	1	Port not opened.
SPS_ERROR_ACCESS_DENIED	5	Error found in a handle to be closed.

Description

This method disconnects a connected port. Properties can be configured even after disconnecting the port.

4.2.4 GetStatus method

GetStatus	Obtaining printer status

Obtains the printer status.

Syntax

Int32 ***GetStatus*** (ref Int32 *Status*)

Parameter

Status : Specifies the buffer used to save a printer status. This is Int32. Low 8 bits are effective.

Table 4-7 and Table 4-8 describe the bits of the printer status.

In addition, *Status* is 0 when failing to obtain the status.

Table 4-7

Bit	Function	Value	
		0	1
0	Paper out	OK	Error
1	Head up	OK	Error
2	VP voltage error	OK	Error
3	Head temperature error	OK	Error
4	DIP switch setting error	OK	Error
5	Battery voltage state	See Table 4-8	
6	Battery voltage state		
7	Reserved	-	Fixed

Table 4-8

Bit 6	Bit 5	Battery voltage state
0	0	8.0 V or higher
0	1	7.5 V to 8.0 V
1	0	7.0 V to 7.5 V
1	1	Lower than 7.0 V

Return value

Table 4-9 shows the values returned by this method.

Table 4-9

Constant	Value	Description
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_INVALID_FUNCTION	1	Port not opened.
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	Failed to obtain a status. (IrDA, LPT) (No printer connected)

Description

Obtains the printer status. The obtained printer status is stored in *Status*. Low 8 bits of the stored data are effective. In addition, if failing to obtain the status, 0 is stored in *Status*.

The library stores the printer status in the internal buffer while on the connection. The printer sends status each time the printer changes its status, the library updates and stores the latest status from the printer automatically. This method stores the retained status data in *Status*.

However, the battery status (Bit 5 and 6) is updated only when other status bits change.

4.2.5 SendText method

SendText — Sending text data

Encodes the specified text data according to the ***CountryCode*** property and sends the data to a printer.

Syntax

Int32 ***SendText*** (String *s*)

Parameter

s: Specifies the text data. This is String type.

The maximum available data size is 3840 bytes.

Return value

Table 4-10 shows the values returned by this method.

Table 4-10

Constant	Value	Description
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_INVALID_FUNCTION	1	Port not opened.
SPS_ERROR_INVALID_PARAMETER	87	0-byte data specified.
		The data size is over 3840 bytes.
		Error occurred in encoding text data.
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	No printer connected. (IrDA)
SPS_ERROR_TIMEOUT	1460	Send or receive timeout.

Description

Sends the text data specified in Parameter *s* to a printer. The maximum available data size is 3840 bytes.

The specified text data is encoded according to the ***CountryCode*** property and sent to a printer. For more details on encoding, see the ***CountryCode*** property

This method does not add a line feed code to the end of text data. Therefore, if no line feed code (CR+LF) is added to the end of the specified data, the data is left in the line buffer of the printer, and part or all of sent text data may not be printed.

See also

CountryCode property

4.2.6 SendBinary method

SendBinary	Sending binary data

Sends a specified binary data to a printer.

Syntax

Int32 ***SendBinary*** (Byte[] *b*)

Parameter

b: Specifies the binary data. This is Byte[].

The maximum available data size is 3840 bytes.

Return value

Table 4-11 shows the values returned by this method.

Table 4-11

Constant	Value	Description
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_INVALID_FUNCTION	1	Port not opened.
SPS_ERROR_INVALID_PARAMETER	87	0-byte data specified.
		The data size is over 3840 bytes.
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	No printer connected. (IrDA)
SPS_ERROR_TIMEOUT	1460	Send or receive timeout.

Description

Sends the binary data specified in Parameter *b*. The maximum available data size is 3840 bytes.

This method sends the specified data to a printer without converting it.

You can use printer functions that these libraries do not support by sending a printer command as binary data with this method. However, it does not support the command to acquire the response from the printer.

4.2.7 SendDataFile method

SendDataFile	Sending data file

Sends a specified text file, bitmap file and binary data file.

Syntax

Int32 ***SendDataFile*** (String *fn*)

Parameter

fn: Specifies a data file name. This is String type.
The maximum file size is 1048576 bytes (1M byte).

Table 4-12 shows the extensions available for data files.

Table 4-12

Extension	File format	Description
.txt	Text file	Encodes text data in a specified file according to the ***CountryCode*** property and sends the data to a printer.
.bmp	Bitmap file	Converts bitmap data in a specified file to a raster bit image and sends it to a printer. Only monochrome bitmap data (two digits) is supported.
.bin .dat	Binary file	Sends binary data in a specified file to a printer without converting it.

Return value

Table 4-13 shows the values returned by this method.

Table 4-13

Constant	Value	Description
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_INVALID_FUNCTION	1	Port not opened.
SPS_ERROR_FILE_NOT_FOUND	2	No specified file found.
SPS_ERROR_OUTOFMEMORY	14	Insufficient memory when loading bitmap data (when specifying a bitmap file).
SPS_ERROR_INVALID_PARAMETER	87	0-byte file specified.
		Over 1048576 bytes file specified.
		The data size in one line exceeds 3840 bytes (when specifying a text file).
		Error occurred when encoding text data (when specifying a text file).
		Unsupported bitmap data specified (when specifying a bitmap file).
		Data of over 65535 dots in the paper width direction specified (when specifying a bitmap file).
		Data of over 4000 dots in the paper feed direction specified (when specifying a bitmap file).
		Data of over 832 dots in both the paper width and paper feed directions specified (when specifying a bitmap file).
SPS_ERROR_DEVICE_NOT_CONNECTED	1167	No printer connected. (IrDA)
SPS_ERROR_TIMEOUT	1460	Send or receive timeout.

Description

A file that specified by Parameter *fn* is determined the type of file based on its extension. This method allows the type of files as a text file, bitmap file and binary file. These files are processed as following.

(1) Processing text file

Data in a text file is encoded by one line according to the ***CountryCode*** property and sent to a printer. The maximum data size available for one line is 3840 bytes. For more details on encoding, see the ***CountryCode*** property.

This method does not add a line feed code to the end of text data. Therefore, if no line feed code (CR+LF) is added to the end of the specified data, the data is left in the line buffer of the printer and part or all of the sent text data may not be printed.

(2) Processing bitmap file

A bitmap file supports only monochrome bitmap data (two digits). In this method, bitmap data in which both the paper width direction and the paper feed direction exceed 832 dots cannot be printed. Bitmap data which exceed 1048576 bytes (1M byte) also cannot be printed.

This method requires enough memory capacity in your system.

In addition, the printing of bitmap may require plenty of time because the .NET Compact Framework 1.0 environment with DPU-S445 NETCF10 takes a long time to deploy an image. We recommend that you use the .NET Compact Framework 2.0 environment with DPU-S445 NETCF20 when you mainly print bitmap data.

If sending of bitmap data is aborted for any reason, a printer falsely recognizes other processes part of the bitmap data until specified bitmap data are successfully received. In this case, abort the status of the printer waiting for bitmap data using the ***Abort*** method.

(3) Processing binary file

A binary file is sent to a printer without being converted. You can use printer functions that these libraries do not support by sending a printer command as binary data with this method. However, the libraries do not support the command to acquire the response from the printer

See also

CountryCode property, ***Abort*** method

4.2.8 Abort method

Abort	Aborting the waiting status of a printer

Aborts the status of a printer waiting for bitmap data.

Syntax

Int32 ***Abort*** ()

Parameter

None

Return value

Table 4-14 shows the values returned by this method.

Table 4-14

Constant	Value	Description
SPS_ERROR_SUCCESS	0	Successfully executed.
SPS_ERROR_INVALID_FUNCTION	1	Port not opened.
SPS_ERROR_TIMEOUT	1460	Send or receive timeout.

Description

If sending of bitmap data is aborted by the ***SendDataFile*** method, a printer does not accept other processes until specified bitmap data are successfully received. (The method or sent data is misinterpreted and recognized as part of the bitmap data.) To solve this situation, use this method and abort the waiting status of the printer. The executing of this method may print unprocessed bitmap data.

See also

SendDataFile method (bitmap file)

4.3 PROPERTY DETAILS

This section provides details on properties.

4.3.1 SendTimeout property

SendTimeout	Time-out period when data is sent

Sets a time-out period when data are sent in msec (milliseconds).

Syntax

SendTimeout

Access Read/Write (R/W)

Data type System.Int32

Default 10000 (msec)(10 sec)

Range 1-60000 (msec)(60 sec)

Description

This property can be configured regardless of whether a printer is connected or not. However, retry the ***Connect*** method after disconnecting a printer using the ***Disconnect*** method once in order to change the setting during connection to the printer because the setting is reflected in the ***Connect*** method to be executed after setting this property.

4.3.2 ReceiveTimeout property

ReceiveTimeout	Time-out period when data is received

Sets a time-out period when data is received in msec (milliseconds).

Syntax

ReceiveTimeout

Access R/W

Data type System.Int32

Default 10000 (msec) (10 sec)

Range 1-60000 (msec) (60 sec)

Description

This property can be configured regardless of whether a printer is connected or not. However, the library does not process promptly with the configured setting, which can be performed after executing ***Connect*** method.
Therefore, retry the ***Connect*** method after disconnecting the printer using the ***Disconnect*** method once in order to change the setting during connection to the printer.

4.3.3 CountryCode property

CountryCode Country code

Sets and obtains a country code as the key to the settings for the international character set for a printer and a character set, and the encode parameter for text data.

Syntax

CountryCode

Access R/W

Data type SII.SII_Printer.SPS_COUNTRY

Default JAPANESE If the .NET Compact Framework supports Japanese

ENGLISH_USA If the .NET Compact Framework does not support Japanese

Range

Table 4-15 shows the constants for country codes configurable through this property.

Table 4-15

Constant (SPS_COUNTRY)	Value	International character set (Printer)	Character set (Printer)	Encode
ENGLISH_USA	0	USA	Enhanced graphics character set	IBM437 *1
FRENCH	1	France	Enhanced graphics character set	IBM437 *1
GERMAN	2	Germany	Enhanced graphics character set	IBM437 *1
ENGLISH_UK	3	UK	Enhanced graphics character set	IBM437 *1
DANISH1	4	Denmark I	Enhanced graphics character set	IBM437 *1
SWEDISH	5	Sweden	Enhanced graphics character set	IBM437 *1
ITALIAN	6	Italy	Enhanced graphics character set	IBM437 *1
SPANISH1	7	Spain	Enhanced graphics character set	IBM437 *1
JAPANESE	8	Japan	Katakana character set	shift_jis *2
NORWEGIAN	9	Norway	Enhanced graphics character set	IBM437 *1
DANISH2	10	Denmark II	Enhanced graphics character set	IBM437 *1
SPANISH2	11	Spain II	Enhanced graphics character set	IBM437 *1
LATIN_AMERICA	12	Latin America	Enhanced graphics character set	IBM437 *1

Description

This property can always be configured. Text data specified with the ***SendText*** method or the ***SendDataFile*** method sets the international character set and character set with this property setting. The text data is converted in any encode format shown in Table 4-15 and sent to the printer. See the technical description for the international character set and character set.

If the .NET Compact Framework installed in your environment does not support Japanese, characters are handled as ENGLISH_USA.

*1: When a country code is not JAPANESE, '?' (3Fh) is printed if the enhanced graphics character set on the printer has no corresponding character for encoding.

*2: When the country code is JAPANESE, JIS level-1 and -2 kanji characters can be printed. However, Shift JIS 8040h-879Ch characters are printed in the format specified by the printer. In addition, characters of Shift JIS ED40h-EDFFh, EE40h-EEFFh, FA40h-FAFFh, FB40h-FBFFh and FC40h-FC4Fh are printed as two-byte spaces (Shift JIS 8140h).

See also

SendText method, ***SendDataFile*** method

4.3.4 Baudrate property

Baudrate	Baud rate

Sets and obtains the baud rate during serial communication.

Syntax

Baudrate

Access R/W

Data type System.Int32

Default 115200 (bps)

Range

Table 4-16 shows the baud rates configurable for this property.

Table 4-16

Value	Description
9600	9600 bps
19200	19200 bps
38400	38400 bps
57600	57600 bps
115200	115200 bps

Description

Sets and obtains the baud rate during serial communication. Though you can use baud rates other than those shown in the table above, this library does not support these baud rates.

This property can be configured regardless of whether a printer is connected or not. However, retry the ***Connect*** method after disconnecting the printer using the ***Disconnect*** method once in order to change the setting during connection to the printer because the setting is reflected in the ***Connect*** method to be executed after setting this property.

4.3.5 Handshake property

Handshake	Flow control

Obtains and sets to the method for controlling the flow during serial communication.

Syntax

Handshake

Access R/W

Data type SII.SII_Printer.SPS_HANDSHAKE

Default Busy

Range

Table 4-17 shows the constants for flow control configurable through this property.

Table 4-17

Constant	Value	Description
XonXoff	1	Xon/Xoff
Busy	2	Hardware

Description

Obtains and sets to the method for controlling the flow during serial communication. Specify ***Connect*** method accordance with this property setting.

This property can be configured regardless of whether a printer is connected or not. However, the library does not process promptly with the configured setting, which can be performed after executing ***Connect*** method.
Therefore, retry the ***Connect*** method after disconnecting the printer using the ***Disconnect*** method once in order to change the setting during connection to the printer.

4.3.6 ByteSize property

ByteSize	Byte size

Obtains the byte size during serial connection.

Syntax

ByteSize

Access Read-only (R)

Data type System.Int32

Default 8 (fixed)

Description

This property obtains the byte size during serial connection.

You can set no value for this property because it is read only.

4.3.7 StopBits property

StopBits	Stop bit size

Obtains the stop bit size during serial connection.

Syntax

StopBits

Access R

Data type System.Int32

Default 1 (fixed)

Description

This property obtains the stop bit size during serial connection.

You can set no value for this property because it is read only.

4.3.8 Parity property

Parity	Parity

Obtains the parity during serial connection.

Syntax

Parity

Access R

Data type System.Int32

Default 0 (fixed) (no parity)

Description

This property obtains the parity during serial connection.

You can set no value for this property because it is read only.

4.3.9 PortName property

PortName	Name of a connected port

Obtains the name of the connected port.

Syntax

PortName

Access R

Data type System.String

Default “” (blank)

Description

This property obtains the name of a connected port which is specified by ***Connect*** method.

You can set no value for this property because it is read only.

4.3.10 PortType property

PortType	Type of a connected port

Obtains the type of the connected port.

Syntax

PortType

Access R

Data type SII.SII_Printer.SPS_PORT

Default PORT_INVALID

Range

Table 4-18 shows the constants for the ports obtainable through this property.

Table 4-18

Constant (SPS_PORT)	Value	Description
PORT_INVALID	0	Disconnected
PORT_SERIAL	1	Connected to COMx:
PORT_IRDA	2	Connected to IrDA
PORT_USB	3	Connected with LPTx:

Description

This property obtains the type of the connected port.

You can set no value for this property because it is read only.

4.3.11 IsOpen property

IsOpen	Connection

Obtains the connection status of this library.

Syntax

IsOpen

Access R

Data type System.Boolean

Default False

Range

Table 4-19 shows the status obtainable through this property.

Table 4-19

Constant	Description
True	Connected (Port:Open)
False	Disconnected (Port:Closed)

Description

This property obtains the status of the connection with a printer in this library.

You can set no value for this property because it is read only.

CHAPTER 5
SAMPLE PROGRAM

This chapter describes the sample programs and the source files provided by these libraries.

5.1 SAMPLES

5.1.1 Sample programs

The sample programs use these libraries on a Microsoft Windows Mobile 5.0-based device.

The sample programs are installed as Sample10.cab and Sample20.cab under the `\Program Files\SII\DPU-S445\sample\cab` folder on the PC where these libraries are installed. (See Figure 3-1)

Table 5-1 shows the versions of the .NET Compact Framework supported by the cab files of these sample programs.

Table 5-1

cab file	Supported .NET Compact Framework
Sample10.cab	.NET Compact Framework 1.0
Sample20.cab	.NET Compact Framework 2.0

.NET Compact Framework 2.0 is not installed on the Windows Mobile 5.0-based device by default. Download and install the Framework from the Microsoft website before using Sample20.cab.

5.1.2 Installing sample programs

To install sample programs on a Microsoft Windows Mobile-based device, copy the cab file (Sample10.cab or Sample20.cab) to any folder on the Windows Mobile-based device and execute the cab file from File Explorer.

Figure 5-1 shows an example in which Sample10.cab is executed and the icon of SII Sample10 is registered in [Programs] on the Windows Mobile-based device.

Figure 5-1

5.1.3 Executing sample programs

Executing a sample program displays the window shown in Figure 5-2. (The same as that of Sample20)

Table 5-2 shows functions.

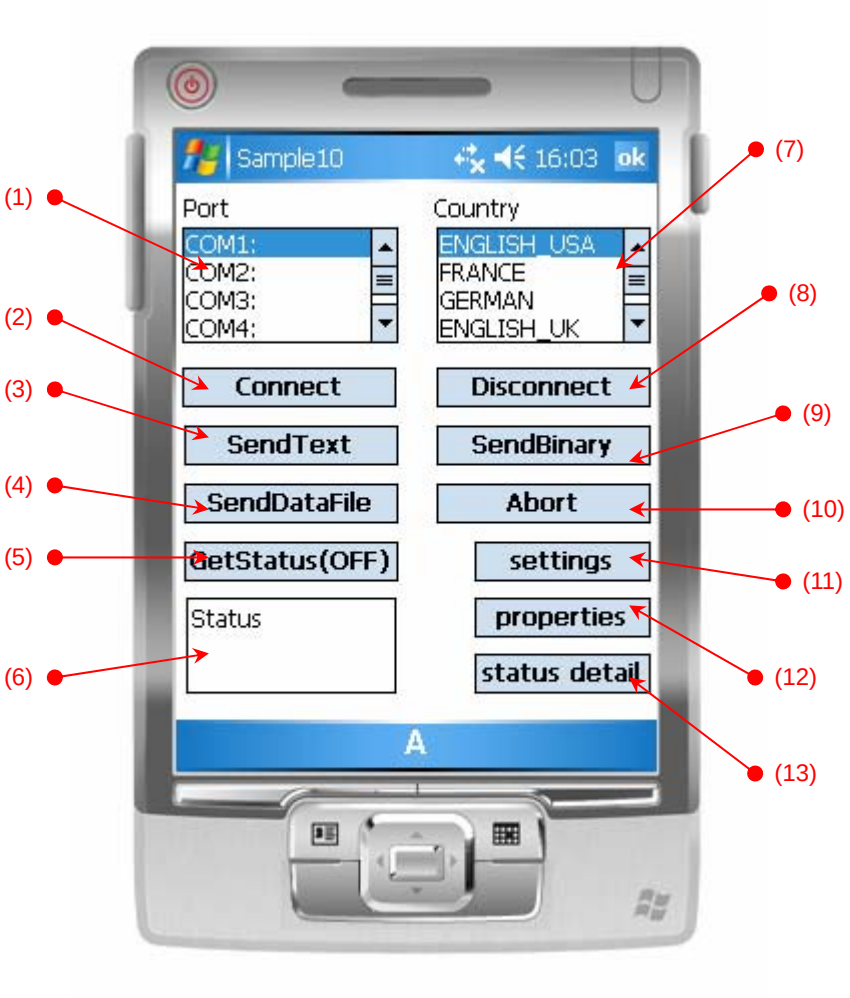


Figure 5-2

Table 5-2

No.	Description	Remarks
(1)	Port selection	Select a parameter of ***Connect*** method among COM1:-COM9:, LPT1:, LPT2: and IR:.
(2)	***Connect*** method	Uses a port selected at (1) as a parameter and executes the ***Connect*** method.
(3)	***SendText*** method	Executes the ***SendText*** method. Displays the text input window as another window.
(4)	***SendDataFile*** method	Executes the ***SendDataFile*** method. Displays the file selection window as another window.
(5)	***GetStatus*** method	Specifies executing the ***GetStatus*** method on a regular basis using the timer or stopping it. (Turning it ON specifies regular execution while turning it OFF specifies stopping it.) An obtained status is displayed at (6). A timer interval for regular execution is 500 msec.
(6)	Result of ***GetStatus*** method	Displays the status when (5) is ***GetStatus*** (ON).
(7)	***CountryCode*** property	Selects the ***CountryCode*** property.
(8)	***Disconnect*** method	Executes the ***Disconnect*** method.
(9)	***SendBinary*** method	Executes the ***SendBinary*** method. Displays the binary data input window as another window.
(10)	***Abort*** method	Executes the ***Abort*** method.
(11)	Settings	Displays the property setting window.
(12)	Properties	Displays the property list window.
(13)	Status Detail	Displays the details of printer statuses obtained with the ***GetStatus*** method in the Status window.

5.1.4 Sample program source file

These libraries provide source files (Visual C# 2005) for sample programs (Sample 10 and 20).

These source files are installed under the `\Program Files\SII\DPU-S445\sample\source` folder on the PC after these libraries are installed. (See Figure 3-1)

5.1.5 Precaution

No guarantee of proper operation and support are provided for sample programs.

Samples programs are subject to change without notice.

CHAPTER 6
DISCLAIMER

We closely monitor the development of this software in order to avoid problems. However, we are not responsible for any damages arising out of the use of this software.