

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

**Brother P-touch Applicable  
Component (b-PAC)  
SDK Specifications Version 2.0**

Brother Industries, Ltd.  
©Copyright Brother Industries, Ltd. 1998-2007

Unauthorized reproduction or duplication is prohibited.



b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

– Table of Contents –

1. Introduction ..... 3

2. During operation ..... 4

3. Creating template files ..... 6

4. BrssCom.Document reference..... 11

5. Sample programs ..... 22

– History of Revisions –

08/02/2001.....Newly created

11/13/2002..... Version 1.0.1 Addition of the bpoContinue flag to DoPrint()

11/13/2002..... Addition of Mobile Printer to the list of printers

03/11/2004..... Version 1.4 Compatible with P-touch Editor Version 4 documents

..... Can specify the driver with the Open method

..... Added the description for the bpoTailCut flag in the description of the DoPrint method

..... Added a description for the bpoChainPrint flag

07/20/2005..... Version 1.5 Compatible with P-touch Editor Version 4.2 documents

09/30/2005..... Version 1.5.2 Addition of Using b-PAC on .NET Framework

..... Addition of VB.NET samples

10/01/2007..... Version 2.0 Compatible with P-touch Editor Version 5.0 documents

---

Microsoft®, Windows®, Visual Basic®, Visual C++®, and ActiveX® are registered trademarks of Microsoft Corporation in the United States and/or other countries. All other company and product names used with this software are trademarks or registered trademarks of their respective companies.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

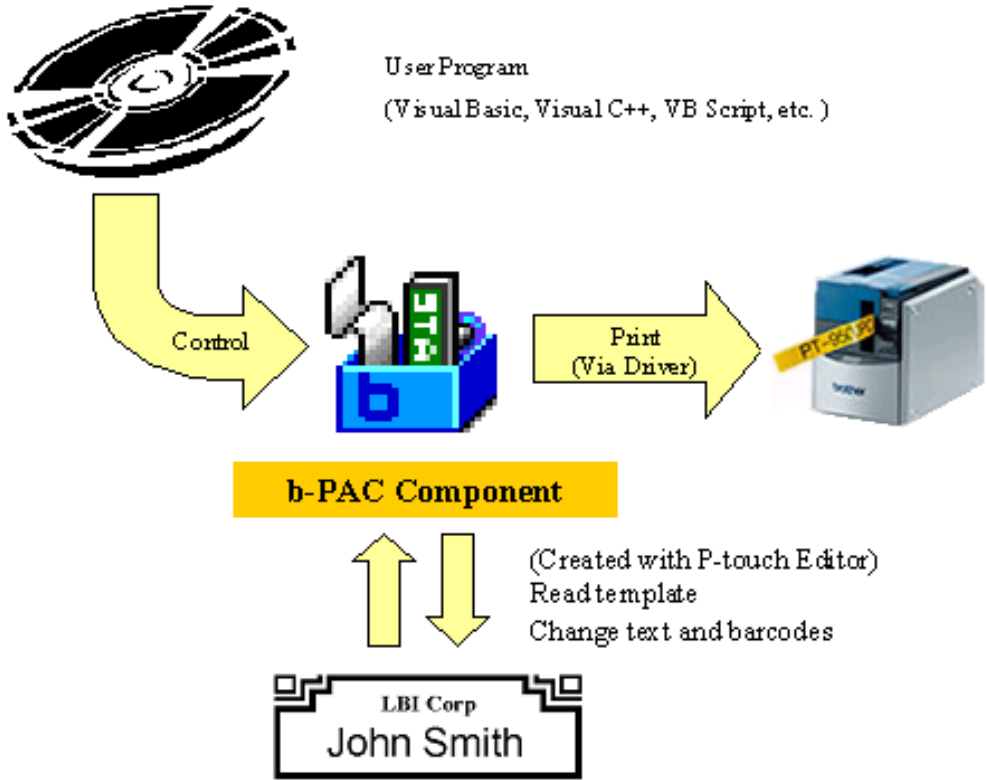
# 1. Introduction

These specifications include explanations and references for the SDK (Software Development Kit) and the **Brother P-touch Applicable Component** (hereafter, simply referred to as "b-PAC". The b-PAC provides an automation function for using P-touch® Editor document files (\*.lbl,\*.lbx) as templates and for outputting them from the user's program to a P-touch series printer without using the Editor.

## 1.1. What is b-PAC?

b-PAC is a component that uses P-touch® Editor document files as templates, replaces text, barcodes and images (illustrations), and outputs to P-touch, Mobile Printer(MW series) and Stampcreator series hardware. (hereafter, simply referred to as "P/M/S series") Therefore, template files created using P-touch® Editor and drivers for the P/M/S series printer used for outputting are required.

b-PAC is packaged as a COM automation server. If an application that can handle COM automation servers, such as Visual Basic® or Visual C++®, or a development environment is used, outputting is possible without using P-touch Editor.



b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

## 2. During operation

### 2.1. User license agreement

In order to use the component and the SDK containing the component, you must agree to the user license agreement during installation. For details on the user license agreement, refer to the file bPACEula.rtf also contained in the same folder where this document was found.

### 2.2. Installation

In order to use the component and the SDK containing the component, run the downloaded installer.

(Note1)

A sample project is included in "Brother b-PAC SDK"; however, the sample project assumes that the templates are located in the folder C:\Program Files\Brother bPAC SDK\Templates.

If Brother b-PAC SDK is installed into a folder other than C:\Program Files\Brother bPAC SDK, copy all files in the installed Templates folder to C:\Program Files\Brother bPAC SDK\Templates.

In order to use the various sample projects, the necessary development environment and execution environment must be installed separately. For more details, refer to 5 [Sample programs](#).

### 2.3. Installed files

When Brother b-PAC SDK is installed, files are copied to the following folders.

- **Application folder (normally C:\Program Files\Brother bPAC SDK\)**
  - \Doc This document, the user license agreement, and a list of redistributable samples
  - \Redist Merge Module for redistribution (BrssCom.msm)
  - \Samples VB, VBA, VBS, and VC sample projects
  - \Templates Template files for use with the sample projects
- **Common files folder (normally C:\Program Files\Common Files\Brother BrssCom)**
  - Component and type library

### 2.4. Uninstallation

To uninstall Brother b-PAC SDK, remove it using "Add/Remove Programs" in the Control Panel.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

## 2.5. Program redistribution

When redistributing programs created using Brother b-PAC SDK, other than the user's program, the following components and files must be installed on the client computer that the redistributed program is installed on.

Brother b-PAC SDK, itself, cannot be redistributed.

### 1. b-PAC component

The easiest and most reliable method for installing the component is to add "BrssCom.msm" from "Redist" in the application folder (normally C:\Program Files\Brother bPAC SDK\) to the install project for Windows® Installer\*. With BrssCom.msm, the component can be used with multiple projects, and the registry and type library of the component are also registered.

(\*[http://msdn2.microsoft.com/en-us/library/8x727h8b\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/8x727h8b(VS.80).aspx))

To install only the component onto a client PC, use the b-PAC client component downloaded from the b-PAC download site.

For detailed operating instructions on the client component, refer to the download site.

### 2. Printer driver of target printer

Set up the driver by running the driver installer from the P-touch Editor CD-ROM, or by downloading the appropriate driver from the Brother Web site (<http://solutions.brother.com>)

Furthermore, if the target driver has been installed together with P-touch Editor onto the client computer, it is not necessary to set up the driver separately.

### 3. Template files

Since b-PAC refers to the templates, the templates created using P-touch Editor are needed. The templates can even be stored on a server accessible by the client computer.

In addition, if the target driver is installed and the template files are available, P-touch Editor is not needed to run b-PAC.

## 2.6. Using b-PAC on .NET Framework

When using b-PAC by managed code in the development environment on and after Visual Studio .NET2003, "Add Reference" is needed.

As for "Add Reference", refer to [http://msdn2.microsoft.com/en-us/library/fwawt96c\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/fwawt96c(VS.80).aspx) or [http://msdn2.microsoft.com/en-us/library/cwys3b23\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/cwys3b23(VS.80).aspx) .

Specify "C:\Program Files\Common Files\Brother BrssCom\BrssCom.tlb" as a tlb file.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

### 3. Creating template files

#### 3.1. What are template files?

In order to use b-PAC, template files must be created using P-touch® Editor Version 3.x/4.x/5.0. With P-touch® Editor, it is not necessary to save template files in a special format; they can be saved as a normal layout file (\*.lbl,lbx).

b-PAC outputs the template file (=layout file) with the same size (width and length) that was set using the printer driver that was set. Therefore, regardless of the current printer settings, you can output using the desired driver and label size.

(\*Note) Also refer to the "[Limitations on b- PAC templates](#)".

#### 3.2. Creating a template

##### 3.2.1. Procedure

1. Start up P-touch Editor Version 3.x/4. Before continuing, select [File] – [Printer Setup], and check that the currently selected driver is the one that you wish to create a template file for. If a different driver is selected, select the desired driver, click the [Set as Default] button, and then click the [OK] button.
2. Specify the label size using the settings in the Page Properties.
3. Enter the desired label data. Observe the precautions described below when adding each type of object to the layout.
4. After creating the layout is finished, save the layout file.

##### 3.2.2. Text objects

- Single text objects should consist of complete meaningful units. For example, a long address that would extend over two lines should be entered at a single multi-line object, instead of two separate objects.
- The font, and character size and style within each text object should be kept to only one type. Each character can be formatted differently; however, during output, all characters will be changed to the formatting specified for the first character.
- Text option settings (When a template is created with P-touch Editor Version 3.x)  
Since certain characteristics of the template text will change, the text alignment and the merge option settings (in the Text Properties) should be used so that the position of the text does not change, even if the width of the text object does. The merge option setting can be used as described below.

**Auto:** This option should be used when there is extra space around the text, but the font size and height-to-width ratio should be maintained without cutting off some of the text, or when the text data in the object is small enough so that the size of the object can be reduced. However, with b-PAC, this option produces the same result as "Clip".

**Clip:** This option is recommended for templates used with b-PAC. If this option is selected and the number of characters is reduced, the font size and height-to-width ratio are maintained without changing the size of the text object. If the number of characters is increased and the size of the text object is enlarged, the font size is reduced without changing the height-to-width ratio.

**Fit:** Select this option to automatically adjust the size of the text to fit the size of the object. With this option, the font size and height-to-width ratio change depending on the number of characters.

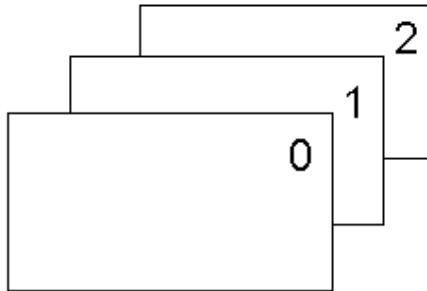
- Text option settings (When a template is created with P-touch Editor Version 4.x)

With the Version 4 format, the alignment of characters in a text object is specified using the Alignment and Text Options settings. Unlike with the Version 3.x format, operations where the SetText ( ) method is specified with b-PAC and text is added or removed conforms with the operations of P-touch Editor Version 4. However, with the “Fixed Frame Size” setting (reducing the text to display all of it while maintaining the height-to-width ratio), the positions of the lines remain fixed even if text is added or removed, and the text alignment position when text is added is “Top” (align at the top), “Center” (align on the center) and “Bottom” (align on the baseline).

- With the Auto and Clip options, the normal font size will have the standard height-to-width ratio (ratio immediately after changing the font size).
- Text objects are not enlarged or reduced linearly, as opposed to graphics objects, etc. For this reason, if text is positioned inside a graphics object with a similar shape, changing the size may result in the text becoming larger than the graphics object. It is recommended not to use excessive precision, but rather to provide extra space when positioning objects.

**3.2.3. Indexing text**

- When indexed with a method such as SetText(), objects of the same type are arranged in the Z order, starting from the front. When indexed, the front object becomes 0, and the remaining become 1, 2, etc.  
 To check the Z order with P-touch® Editor, select the [Text Box] command. With Text Box, objects containing text are displayed in order, starting from the front. Therefore, after all text has been entered, start with the text object (or Arrange Text object) entered last and select [Layout] – [Order] – [Bring to Front] to bring the object to the front. (Repeat this operation on all entered text objects to reverse their order.)



In addition, if a text object contains multiple lines of text, the lines are numbered starting from the top. The order within an object cannot be changed. (Each single line of text is indexed.)

With the Version 4 format, if “Text Options” for text is set to “Text Length”, there is one index for the entire object.

With the Version 4 format, text in table objects are also indexed. Text objects in tables are indexed starting from the upper-left cell and ending with the lower-right cell.

0	1	2
3	4	5

If independent text objects are combined with tables containing text objects in cells, the tables and text are first indexed in the Z order, and then, indexes are added in the cell order described above for the table object.

**3.2.4. Graphics objects (illustration)**

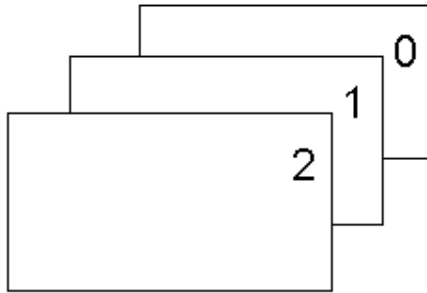
- If graphics objects (illustrations) contained in the template are to be replaced, templates

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

containing smaller images with fewer colors can be loaded more quickly. b-PAC can import Jpeg files; however, when creating templates, it is recommended that images be replaced with the smaller Bmp files.

### 3.2.5. Barcode objects

- Like text objects, barcode objects cannot be resized linearly. For this reason, it is recommended that they be positioned with extra space around them.
- When indexed with a SetBarcodeData() method, objects of the same type are arranged in the Z order, starting from the back. When indexed, the back object becomes 0, and the remaining become 1, 2, etc.  
Therefore, after all barcodes has been entered, start with the barcode object entered first and select [Layout] – [Order] – [Bring to Front] to bring the object to the front. (Repeat this operation on all entered barcode objects to reverse their order.)



### 3.2.6. Other objects

- If text is positioned in a frame, the frame is sent behind the text.

### 3.2.7. Limitations on b-PAC templates

Of the layout files (\*.lbl,\*.lhx) created with P-touch® Editor Version 3.x/4.x/5.0, many can be used as templates with b-PAC. However, the following limitations apply.

- Layout files containing merged database fields cannot be used as templates.
- Clip art, Make Picture objects cannot be displayed with the current version of b-PAC. If you wish to use templates containing these types of objects, contact us for more information.
- Printer for outputting the (template) data

The information for the printer driver used for outputting is saved with the template; therefore, the computer will try to use the set printer driver whenever outputting the template. If the printer driver that was set when the template was created is not installed on the PC that will output the b-PAC data, the following will occur.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

No printer driver has been installed:		The b-PAC component cannot be used.
If a printer driver has been installed, but a P-touch/Mobile Printer/Stampcreator-series (hereafter, referred to as "P/M/S-series") printer driver has not:		The template can be opened, but it cannot be printed.
If a P/M/S-series printer driver has been installed, but it is different than the one used when the template was created:	When only one P/M/S-series driver is installed:	The data will be outputted from the printer for the installed P/M/S-series driver.
	When multiple P/M/S-series drivers have been installed:	The data will be outputted from the P/M/S-series printer with the driver name that is first alphabetically. In this case, the template is adjusted to match the printer driver installed on the PC.

### 3.2.8. Sample templates included with SDK

The print area and printer for the templates in the b-PAC SDK template folder (normally C:\Program Files\Brother bPAC SDK\Templates) are have been set for the PT-9200PC.

If output is possible with P/M/S-series printer drivers other than the PT-9200PC, use P-touch Editor to open a layout file (\*.lbl) from the above-mentioned template folder, select [File] - [Printer Setup], select the printer that will be used to output the data, adjust the layout if it extends out of the print area, and then save the file.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

### 3.3. Checking a template

After creating the template file, open it and check that it is displayed correctly. Below is a checklist that should be referred to when creating a template.

#### 3.3.1. Overall layout

- With label templates, do the objects have enough extra space within the print area?
- With stamp templates, do all objects fit within the print area? (However, special layouts containing many Arrange Text objects and clip art are excluded.)
- Have all unnecessary objects been removed from the layout? (Check by holding down the Tab key to select each object.)
- With no object selected, select the [Text Box] command. Is the order in which the text appears the desired order?

#### 3.3.2. Text

- Except when the Fit option is selected, does the font of all text appear in the original height-to-width ratio (especially when the Auto option is selected)?
- If the Clip option is selected, is the text object located so that it will not hide other objects (lines, etc.), does it provide for characters being added, and is it adjusted to a size with enough extra space?
- Have no unnecessary spaces been added to the text? Instead, is it not possible to adjust the character spacing?
- Is any necessary character formatting lost when the text is displayed using the [Text Box] command (when character formatting is applied)?

#### 3.3.3. Barcode

- Is there enough space around the barcode object to prevent it from extending beyond the print area if the length changes due to the barcode data or from covering other objects on either side (when the barcode is horizontally positioned)?

#### 3.3.4. Others

- Display the layout at a zoom ratio of 1:1. Check how the objects overlap. Be particularly careful when grouping rectangles and straight lines and when drawing lines. In addition, if a bitmap is used in the template, check how it will be reproduced.
- Display the layout at a zoom ratio of 100%. Are there no missing parts in the objects and is the balance of the objects the same as when the layout is displayed normally?

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

## 4. BrssCom.Document reference

### 4.1. Interface name

The name of the interface for b-PAC is “BrssCom.Document”.

### 4.2. List of methods

Method	Summary
<a href="#">Open()</a>	Opens the lbl file at the specified path (UNC).
<a href="#">Close()</a>	Closes the document.
<a href="#">Save()</a>	Saves the document.
<a href="#">Export()</a>	Outputs the file in the specified format (LBL, LBI or BMP). For the LBI and BMP formats, the bitmap resolution can be specified.
<a href="#">GetSheetSize()</a>	Retrieves the string of characters that define the paper size.
<a href="#">GetTextCount()</a>	Returns with the number of lines of text in the document.
<a href="#">GetTextIndex ()</a>	Returns with an index for the text consistent with the field name.
<a href="#">GetText()</a>	Retrieves the text data in the specified line.
<a href="#">GetFontInfo()</a>	Retrieves the font for the specified line.
<a href="#">SetText()</a>	Sets the text data for the specified line.
<a href="#">SetFontInfo()</a>	Sets the font for the specified line.
<a href="#">SetBarcodeData()</a>	Sets and updates the barcode data.
<a href="#">ReplaceImageFile()</a>	Replaces the graphics object.
<a href="#">DoPrint()</a>	Prints the document.

#### 4.2.1. Open()

**boolean Open(BSTR *szFilePath*)**

**[Arguments]**

*szFilePath* Path (UNC) to the document file

**[Returned value]**

“0” is returned if the open failed; a value other than “0” is returned if it succeeded.

**[Description]**

Opens the lbl file at the specified path (UNC).

By adding “/Ddriver\_name” (ex. /DBrother PT-9500PC) to the path for the template file, instead of the driver being stored in the template file, the driver is explicitly specified by the method from the beginning. In this way, one template can be filtered to multiple printers for printing. However, if the specified driver does not exist, the output is performed according to the template output rules in [“3.2.7 Limitations on b-PAC templates”](#).

**[Usage example of the Open method with Visual Basic]**

- Normally called  
Open(“C:\Program Files\Brother bPAC SDK\Templates\NamePlate1.lbl”)
- Called with the output printer driver specified  
Open(“C:\Program Files\Brother bPAC SDK\Templates\NamePlate1.lbl /DBrother PT-9500PC”)

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

#### 4.2.2. Close ()

##### boolean Close()

##### [Returned value]

“0” is returned if the document failed to close; a value other than “0” is returned if it succeeded.

##### [Description]

Closes the file opened with Open.

#### 4.2.3. Save ()

##### boolean Save()

##### [Returned value]

“0” is returned if saving the file failed; a value other than “0” is returned if it succeeded.

##### [Description]

Saves the file opened with Open.

#### 4.2.4. Export()

##### boolean Export(long *FileType*, BSTR *szFilePath*, long *dpi*)

##### [Arguments]

<i>FileType</i>	Type of file to be exported as (0: LBL format; 1: LBI format <sup>1</sup> ; 2: BMP (monochrome) format)
<i>szFilePath</i>	Path (UNC) of the exported file
<i>dpi</i>	Resolution when exporting as a bitmap; valid only if <i>FileType</i> is the LBI or BMP format Specify the resolution of the output device (72 or 96 for a display; 360 for output to the PT-9200PC/9200DX, 600 for output to the SC-2000)

##### [Returned value]

“0” is returned if the export of the file failed; a value other than “0” is returned if it succeeded.

##### [Description]

Exports the file in the format specified with *FileType*.  
If *FileType* is the LBI or BMP format, specify the resolution with *dpi*.

#### 4.2.5. GetSheetSize ()

##### VARIANT GetSheetSize(short *vt*)

##### [Arguments]

<i>vt</i>	<b>VARIANT</b> format for the size information; Normally 8 is specified.
-----------	--

##### [Returned value]

Returns the size information in the **VARIANT** form. For size 1850, “1850” is set for the VT\_BSTR form.

##### [Description]

Retrieves the paper size for the current document in the **VARIANT** form.

<sup>1</sup> In order to increase the portability of the fonts, create a binary file with all proprietary P-touch (Stamp) Editor objects converted to bitmaps. The paper size information is maintained.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

#### 4.2.6. **GetTextCount ()**

**long GetTextCount()**

**[Returned value]**

Number of lines in the current document where text can be entered

**[Description]**

Returns the number of lines in the current document where text can be entered.

#### 4.2.7. **GetTextIndex ()**

**long GetTextIndex (BSTR *szFieldName*)**

**[Arguments]**

*szFieldName* Field name (template text) for the text whose index is to be retrieved

**[Returned value]**

-1: No corresponding lines of text; 0, etc.: Index of the text corresponding to the field name

**[Description]**

Returns the index for text with the specified field name (template text). By specifying a field name in the template, the text can simply be entered.

If there are multiple lines of text with the same field name (template text) in the same template file, the index for the first text line is returned.

Refer to [3.2.3 Indexing text](#) for details on the rules for numbering indexes for text lines.

#### 4.2.8. **GetText ()**

**boolean GetText(long *nIndex*, BSTR\* *lpszText*)**

**[Arguments]**

*nIndex* Index (from 0) for the text line to be retrieved

*lpszText* Pointer to the buffer where the text is to be retrieved

**[Returned value]**

“0” is returned if retrieving the text in the specified line failed; a value other than “0” is returned if it succeeded.

**[Description]**

Retrieves the text data in the specified line.

Warning: You cannot use this method from VBScript or JScript.

**4.2.9. GetFontInfo ()**  
**boolean GetFontInfo(long nIndex, BSTR\* lpszFontInfo)**

**[Arguments]**

*nIndex*            Index (from 0) for the line whose font information is to be retrieved  
*lpszFontInfo*    Pointer to the text of the font information to be retrieved; Refer to [Description] for details.

**[Returned value]**

“0” is returned if retrieving the font information for the specified line failed; a value other than “0” is returned if it succeeded.

**[Description]**

Retrieves the font information for the specified line. The text specified by *lpszFontInfo* is separated by “,” (commas), and each token has the meaning described below.

Index	Name	Value
0	Font name	<i>Name of font</i>
1	Orientation	0: Horizontally written; 1: Vertically written (invalid with SetFontInfo())
2	Bold specification	0: Normal; 1: Bold
3	Italic specification	0: Normal; 1: Italic
4	Underline specification	0: None; 1: Applied
5	Strikeout specification	0: None; 1: Applied
6	Maximum point size	1~: Maximum point size (in 0.1 point units)
7	Character style	0: None; 1: Shadow Light; 2: Shadow 3: Horizontal; 4: Outline 5: Surround; 6: Frame Out

Example: “Arial, 0, 1, 1, 0, 0,120,0” indicates the Arial font, horizontally written, bolded, italicized, maximum 12 points, with no character styles.

Warning: You cannot use this method from VBScript or JScript.

**4.2.10. SetText ()**  
**boolean SetText(long nIndex, BSTR szText)**

**[Arguments]**

*nIndex*            Index (from 0) for the text line to be set  
*szText*            Text to be set

**[Returned value]**

“0” is returned if setting the text of the specified line failed; a value other than “0” is returned if it succeeded.

**[Description]**

Updates and sets the text for the specified line.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

**4.2.11. SetFontInfo ()**  
**boolean SetFontInfo(long nIndex, BSTR szFontInfo)**

**[Arguments]**

*nIndex*            Index (from 0) for the line whose font information is to be set  
*szFontInfo*        Text of the font information to be set; Refer to **"GetFontInfo ()"** for details on the settings.

**[Returned value]**

"0" is returned if setting the font information for the specified line failed; a value other than "0" is returned if it succeeded.

**[Description]**

Updates and sets the font information for the specified line. For details on the font information parameters to be set, refer to **"GetFontInfo ()"**. If omitted (or if set to "-1"), the current font information for that parameter is adopted.

**4.2.12. SetBarcodeData ()**  
**boolean SetBarcodeData(long nIndex, BSTR szData)**

**[Arguments]**

*nIndex*            Index (from 0) for the barcode object  
*szData*            Barcode data text to be set

**[Returned value]**

"0" is returned if setting the barcode failed; a value other than "0" is returned if it succeeded.

**[Description]**

Sets and updates the barcode data for the specified barcode.

Refer to [3.2.5 Barcode objects](#) for details on the rules for numbering indexes for barcodes

[szData] (for only CODE128/EAN128 barcodes)

To insert FNC1 into the data, "\a" should be specified. E.g. when you set "1234"+FNC1+"567"+FNC1 with VBS, write

ObjDoc.SetBarcodeData 0, "1234\a567\a"

When you use VC etc, write "\\a".

The rules of the control codes are;

FNC1	"\a"				
FNC2	"\b"				
FNC3	"\c"				
FNC4	"\d"				
NUL	"\0"	VT	"\B"	SYN	"\M"
SOH	"\1"	FF	"\C"	ETB	"\N"
STX	"\2"	CR	"\D"	CAN	"\O"
ETX	"\3"	SO	"\E"	EM	"\P"
EOT	"\4"	SI	"\F"	SUB	"\Q"
ENQ	"\5"	DLE	"\G"	ESC	"\R"
ACK	"\6"	DC1	"\H"	FS	"\S"
BEL	"\7"	DC2	"\I"	GS	"\T"
BS	"\8"	DC3	"\J"	RS	"\U"
HT	"\9"	DC4	"\K"	US	"\V"
LF	"\A"	NAC	"\L"	DEL	"\W"

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

#### 4.2.13. ReplacelImageFile()

**boolean ReplacelImageFile(long nIndex, BSTR szFilePath, long dwFlag)**

**[Arguments]**

*nIndex*            Index (from 0) for the graphics object to be set  
*szFilePath*        Path of the image file, which should replace the graphics object  
*dwFlag*             Flag for replacing images (Specify "0" or "4".)

**[Returned value]**

"0" is returned if replacing the image failed; a value other than "0" is returned if it succeeded.

**[Description]**

Replaces the specified graphics object with the graphics file specified by *szFilePath*. The formats for the graphics file that can be specified by *szFilePath* are BMP or uncompressed TIFF, WMF, EMF, ICO or JPEG.

When "0" is specified for *dwFlag*, the size of image object is fitted with the original size of the template  
When "4" is specified for *dwFlag*, the aspect ratio of the image object is kept in the template.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

**4.2.14. DoPrint()**  
**void DoPrint(long dwOption, BSTR szOption)**

**[Arguments]**

*dwOption*      Printing options; 0: Adopts the settings of the current driver; Others: Refer to "[Descriptions]".

*szOption*      Printing parameter Specify the printing parameters with strings of characters separated by "," (commas); Refer to [Description] for details.

**[Description]**

Prints the current document. DoPrint() is valid only when printing to a Brother P-touch/Mobile Printer(MW series) /Stampcreator-series printer. It is not used with other printers.

➤ Printing option: *dwOption*

If the driver's print option settings are not to be adopted and the printing options are to be set using DoPrint, specify the settings by combining the following flags in *dwOption*. If the settings are to be adopted, set *dwOption* to "0".

(With Visual Basic®, when the flag is specified with a direct numeral, not an enumerating value that begins with bpo, the half cut flag 0x200, for example, is specified as &H200.)

Driver	Flag	Description
<b>Label printer (P-touch)</b>	bpoAutoCut (0x1)	Performs an auto cut.* Valid only with models supporting the auto cut function.
	bpoCutPause (0x2)	Performs a cut pause or prints cut marks. Valid only with models not supporting the auto cut function.
	bpoHalfCut (0x200)	Performs a half cut.*
	bpoChainPrint(0x400)	Performs "Chain Printing"*
	bpoTailCut(0x800)	The tailcut function is specific to the PT-9500PC. Print onto the small piece of tape that is normally cut off at the start of every label, by splitting your label design into two sections – the left side of the label (23mm) and the right side of the label (for the remainder). When these are printed in succession, the smaller 23mm part of the label prints onto the tape that is normally cut-off.*
	bpoCutAtEnd(0x04000000)	Performs "Cut at end"*
	bpoNoCut (0x10000000)	(The models which have any cut functions) Do not cut.
	bpoMirroring (0x4)	Prints a mirror image.*
	bpoQuality (0x00010000)	Gives priority to print quality*
	bpoColor (0x8)	Prints in color.*
bpoMono (0x10000000)	(The models which have color print function) Print in monochrome.	
bpoContinue(0x40000000)	Combines with printing for the following DoPrint() so that it is a single print job. As a result, when the next DoPrints are called up, the front margins are not outputted. <i>Refer to the "Precautions and limitations for use of the bpoContinue flag" described below.</i>	

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

<b>Mobile Printer (MW Series)</b>	bpoContinue(0x40000000)	Combines with printing for the following DoPrint() so that it is a single print job. As a result, multiple labels can be printed together on a single sheet for precut labels. <i>Refer to the "Precautions and limitations for use of the bpoContinue flag" described below.</i>
<b>Stamp (Stampcreator)</b>	bpoStamp (0x80)	Engraves a stamp. If this flag is set to "OFF", an ID label is printed.
	bpoldLabel (0x10000000)	Print the ID label.

\*Note: Valid only for models supporting the function. The setting is invalid with models not supporting the function, even if the flag is set.

➤ Printing parameter: *szOption*

Specify the printing parameters *szOption* using text separated by “,” (commas). Each token has the meaning described below.

Index	Name	Description
0	Name of document being printed	Specify the document name that will appear in the printer display. If not specified or if “0” is specified, “b-PAC” will appear in the displayed.
1	Number of copies	Specify the number of copies. If not specified, the number of copies is normally “1”. With stamps, the number of copies setting is invalid.

Example: “MyDoc, 2” indicates that “MyDoc” will appear in the printer display, and 2 copies will be printed.

[Quick reference matrix of examples of cut options](#)

**[Precautions and limitations for use of the bpoContinue flag]**

- ✓ Each time the normal DoPrint method is called up, a print job is created. However, by specifying a bpoContinue flag, the end of the print job is postponed until the next one is called up. In addition, if the same flag is also specified for the next DoPrint, the end of the print job continues to be postponed.
- ✓ If the bpoContinue flag is set, other specified flags for the printing options (dwOption) and specified printing parameters (szOption) other than the number of copies are valid for the first DoPrint method called up, but are ignored for following DoPrint methods called up until the print job is finished.
- ✓ Once the bpoContinue flag is set in a DoPrint method, the print job can be finished (=printing actually begins) in any of the following ways.
  - a). Not setting a bpoContinue flag when calling up one of the following DoPrint methods
  - b). Calling up the Close method
  - c). Specifying and calling up with the Open method a template with a different printer driver and paper size (label width with label machines)
  - d). Releasing the b-PAC object
- ✓ After setting a bpoContinue flag and calling up the DoPrint method, it will not enter the process where it waits for an end-user event, such as data input or selection, until the print job is finished (=printing actually begins) with one of the three ways (a, b or c) mentioned above.
- ✓ The maximum number of times a DoPrint method with the bpoContinue flag specified can be called up (number of pages that can be output during one print job) depends on the maximum number of pages that each printer can print or on the computer's resource.

Effect of “bpoContinue” (It is efficient when multiple “open” are used continuously.)

	Enumerating value that begins with bpo	Direct numeral	Print Result
“Auto Cut” without “bpoContinue”	ObjDoc.Open (***.lbl) ObjDoc.DoPrint bpoAutoCut, “0” ObjDoc.Open (****.lbl) ObjDoc.DoPrint bpoAutoCut, “0”	ObjDoc.Open (***.lbl) ObjDoc.DoPrint &H1, “0” ObjDoc.Open (****.lbl) ObjDoc.DoPrint &H1, “0”	
“Auto Cut” with “bpoContinue”	ObjDoc.Open (***.lbl) ObjDoc.DoPrint bpoAutoCut+bpoContinue, “0” ObjDoc.Open (****.lbl) ObjDoc.DoPrint bpoAutoCut, “0”	ObjDoc.Open (***.lbl) ObjDoc.DoPrint &H40000001, “0” ObjDoc.Open (****.lbl) ObjDoc.DoPrint &H1, “0”	

**[Detailed explanation of TailCut function]**

This function is only applicable to the PT-9500PC.

To maximize tape usage, print your label in two sections. The left section (23mm) prints onto the tape that is normally cut off at the start of every label.

Using the following label as an example:



We would create two label templates – one of 23mm that is the left side of the label (LEFT.LBL):



and one containing the remainder of the label (RIGHT.LBL).



The LEFT.LBL label was printed during the last print job and is inside the PT-9500PC.

The RIGHT.LBL is sent to the printer, along with the “ChainPrint” option (telling the printer not to feed the tape out after printing). Then, the smaller LEFT.LBL is sent to the printer, along with the “ChainPrint + TailCut” option. This tells the printer to print the label, but cut the previous label printed, and leave the LEFT.LBL label in the printer ready for the next print job.



An example of VBScript to print these two separate labels next to each other, thereby avoiding any margins before the labels.

```

bRet = ObjDoc.Open("RIGHT.LBL")
If (bRet <> False) Then
    ObjDoc.DoPrint &h0400, "0" 'ChainPrint
End If
bRet = ObjDoc.Open("LEFT.LBL")
If (bRet <> False) Then
    ObjDoc.DoPrint &h0C00, "0" 'TailCut+ChainPrint
End If

```

By repeating the above script, the two labels are combined into one label, with no tape margin.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

**[Usage example of the DoPrint method with Visual Basic®]**

- Prints one copy with the print option settings of the current driver

```
...
ObjDoc.DoPrint 0, "0"
```

- Prints one copy without cutting between the labels

```
ObjDoc.DoPrint bpoNoCut, "0"
or
ObjDoc.DoPrint &H10000000, "0"
```

- Prints three copies with the print option settings of the current driver

```
...
ObjDoc.DoPrint 0, "b-PAC Sample,3"
```

- Example of batch printing of multiple labels using the bpoContinue flag

```
'Prints five labels as one print job (for the models with half cut function)
...
iTotal = 5
nIndex = ObjDoc.GetTextIndex("Name")
For iCount = 1 To iTotal
    strCount = "Count = " + CStr(iCount)
    ObjDoc.SetText nIndex, strCount
    If iCount < iTotal Then
        'Continues the print job and prints with half cuts
        ObjDoc.DoPrint &H40000200, "test,1"
    Else
        'Last print (end of the job)
        ObjDoc.DoPrint 0, "0"
    End If
Next
```

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

## 5. Sample programs

The sample projects assume that the templates are located in the folder C:\Program Files\Brother bPAC SDK\Templates. If Brother b-PAC SDK is copied into any other folder, refer to the precautions in [Installation](#). In addition, this limitation only applies to the sample projects, not to the b-PAC component.

### 5.1. Visual Basic® (VB folder)

<b>NamePit</b>	Summary	Application for creating nameplates Enter the name and company to create nameplates. Select one of the following two templates: simple and framed.
	Methods used	Open(), GetTextIndex(), SetText(), DoPrint()
	Operable development environment	Microsoft® Visual Basic® Version 6.0 SP4

The following shows the b-PAC methods used in the **NamePit** sample.

```
Public Sub DoPrint(strPath As String, strExport As String)

    ' Creates the b-PAC object
    Dim ObjDoc As BrssCom.Document
    Set ObjDoc = CreateObject("BrssCom.Document")

    ' Opens the template
    If (ObjDoc.Open(strPath) <> False) Then

        ' Searches for text from the field names, and replaces the text
        Dim nIndex As Integer
        nIndex = ObjDoc.GetTextIndex("Company")
        ObjDoc.SetText nIndex, edPosition.Text
        nIndex = ObjDoc.GetTextIndex("Name")
        ObjDoc.SetText nIndex, edName.Text

        ' Printing or exporting
        If (strExport = "") Then
            ObjDoc.DoPrint 0, "0"           ' Print
        Else
            ObjDoc.Export 2, strExport, 180 ' Export
        End If
    End If
    Set ObjDoc = Nothing
End Sub
```

## 5.2. Visual C++<sup>®</sup> (VC folder)

<b>NamePlt</b>	Summary	Application for creating nameplates Enter the name and department to create nameplates. Select one of the following two templates: simple and framed. Visual C++ version of the Visual Basic <sup>®</sup> sample <b>NamePlt</b>
	Methods used	Open(), GetTextIndex(), SetText(), DoPrint()
	Operable development environment	Microsoft <sup>®</sup> Visual C++ <sup>®</sup> Version 6.0 SP4

## 5.3. Visual Basic<sup>®</sup> for Application (VBA folder)

<b>ShipForm.xls</b>	Summary	Microsoft <sup>®</sup> Excel application for creating shipping labels. Using an interactive interface, a shipping label is created from text entered in the cells. In addition, the fonts for the text can be changed.
	Methods used	Open(), SetText(), DoPrint(), GetTextCount(), Save(), Close(), SetFontInfo(), Export()
	Operable development environment	Microsoft <sup>®</sup> Excel 97 or, 2000

<b>NamePlt.mdb</b>	Summary	Microsoft <sup>®</sup> Access application for creating nameplates. References the database record data, and prints a nameplate.
	Methods used	Open(), GetTextIndex(), SetText(), DoPrint()
	Operable development environment	Microsoft <sup>®</sup> Access 97 With Access 2000, files can be converted.

<b>ShopLbl.mdb</b>	Summary	Microsoft <sup>®</sup> Access application for creating a price labels References the database record data, and prints a a price label containing the barcode data.
	Methods used	Open(), GetTextIndex(), SetText(), DoPrint(), SetBarcodeData()
	Operable development environment	Microsoft <sup>®</sup> Access 97 With Access 2000, files can be converted.

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

#### 5.4. VB Script (for Windows Scripting Host) (VBS folder)

<b>NamePrn.vbs</b>	Summary	Script for creating nameplates Script printing version of the Visual Basic® sample <b>NamePlt</b>
	Methods used	Open(), SetText(), DoPrint()
	Operable development environment	Microsoft® Windows® Script Host Version 1.0

#### 5.5. VB Script (for Internet Explorer) (VBS folder)

In order to use the following sample, Internet Explorer 4.0 or higher is required.

<b>VBS_NamePlate.html</b>	Summary	HTML page for creating nameplates Enter the name and department to create nameplates. Select one of the following two templates: simple and framed. HTML version of the Visual Basic® sample <b>NamePlt</b>
	Methods used	Open(), GetTextIndex(), SetText(), Export(), DoPrint()

<b>VBS_ChangeImage.html</b>	Summary	HTML page for creating labels containing images Specify the image and enter the name to create the label. Optional bitmap files can be specified for the graphics.
	Methods used	Open(), SetText(), Export(), DoPrint(), ReplaceImageFile()

<b>VBS_DB_Bcd.html</b>	Summary	HTML page for accessing a database and printing labels containing the barcode data and name for the specified item First, create a Microsoft® Access MDB file corresponding to the items and barcode data. While holding down the print or preview buttons, access the database via ADO, retrieve the barcode data corresponding to the item, and then print after the component has performed the replacement with the actual barcode.
	Methods used	Open(), SetText(), Export(), DoPrint(), SetBarcodeData()
	Note	In order to access the database using ADO, the MDAC component (mdac_typ.exe) must be downloaded from the following page, and installed. <a href="http://www.microsoft.com/data/download.htm">http://www.microsoft.com/data/download.htm</a> (This component is already installed on Windows® 2000/XP and Millennium Edition.)

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

## 5.6. Console (command prompt) application (Console folder)

<b>ConsoleLabel</b>	Summary	A sample application that uses the b-PAC component from the console (command prompt) application and outputs barcodes. For an explanation of the program, refer to the following.
	Methods used	Open(), SetText(), SetBarcodeData(), DoPrint()
	Operable development environment	Microsoft® Visual C++® Version 6.0 SP4 Microsoft® Windows® Script Host Version 1.0

### Description

#### **ConsoleLabel.c**

Source of the console program created with C.

The VB Script engine wscript is called from the function system() of the C run-time library. In addition, the script file name (BcdLabel.vbs) and the two arguments transferred to the script file are specified as arguments of wscript. When simplified, it appears as shown below.

```
system("wscript script-file-name parameter1 parameter2");
```

#### **BcdLabel.vbs**

The b-PAC component is called with the VB Script file, called up from ConsoleLabel.c, and the barcode label is printed.

This script file uses two arguments. The first argument is used as the item name, and the second argument is used as the code for the item that will be made into the barcode.

## 5.7. VB.NET(VBNET folder)

<b>TestPrint</b>	Summary	Application for printing templates Optional template files can be selected and printed.
	Methods used	Open(), DoPrint()
	Operable development environment	Microsoft® Visual Basic® .NET 2003


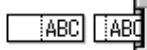
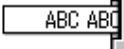
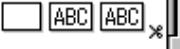


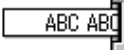

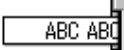

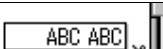
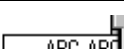
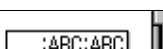
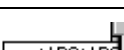

## APPENDIX

### Quick reference matrix of examples of cut options

The driver's print option settings are not to be adopted and the printing options are to be set using DoPrint.

(Available models may differ by countries/regions.)

Model	Cut Option						Print Result
	No Cut	Auto Cut	Half Cut	Chain Print	Cut Mark	Cut at End	
	bpoNocut	bpoAutoCut	bpoHalfCut	bpoChainPrint	bpoCutPause	bpoCutAtEnd	
	&H10000000	&H1	&H200	&H400	&H2	&H4000000	
PT-9300PC PT-9200DX	Yes	-	-				
	-	Yes	-				
	-	-	Yes				
	-	Yes	Yes				
PT-9500PC PT-9600	Yes	-	-	-			
	-	Yes	-	-			
	-	-	Yes	-			
	-	-	-	Yes			
	-	Yes	Yes	-			PT-9500PC 
	-	Yes	-	Yes			PT-9600 
	-	-	Yes	Yes			

		-	Yes	Yes	Yes			PT-9500PC 
								PT-9600 
PT-2300 PT-2600	Yes	-						
	-	Yes						
PT-18NR PT-24	Yes	-			-			
	-	Yes			-			
	-	-			Yes			
	-	Yes			Yes			
PT-1500PC PT-2420PC	Yes					-		
	-					Yes		
PT-1950	Yes					-		
	-				Yes			
	-					Yes		
	-				Yes	Yes		
QL-55 0	Yes	-						

b-PAC SDK Specifications	Create	8/2/01	B.I.L.	Update	10/01/07	B.I.L.
--------------------------	--------	--------	--------	--------	----------	--------

	-	Yes					
QL-1050 QL-650TD*	Yes	-				-	
	-	Yes				-	
	-	-				Yes	
	-	Yes				Yes	

QL-650TD\* : Firmware is version4.0 or later, Driver is version4.1 or later

Example	Enumerating value that begins with bpo	Direct numeral
"Auto Cut" only	ObjDoc.Open (***.lbl) ObjDoc.DoPrint bpoAutoCut, "0,2"	ObjDoc.Open (***.lbl) ObjDoc.DoPrint &H1, "0,2"
"Auto Cut" and "Half Cut"	ObjDoc.Open (***.lbl) ObjDoc.DoPrint bpoAutoCut+bpoHalfCut, "0,2"	ObjDoc.Open (***.lbl) ObjDoc.DoPrint &H201, "0,2"