Special ID Setting Tool

General Overview

- Be sure you have read and understood this document before using this software.
- We recommend that you keep this document nearby for future reference.
- The contents of this document and the specifications of this product are subject to change without notice.
- Brother reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical and other errors relating to the publications.
- The screen images in this document may differ depending on your OS or printer.

ESSG00010

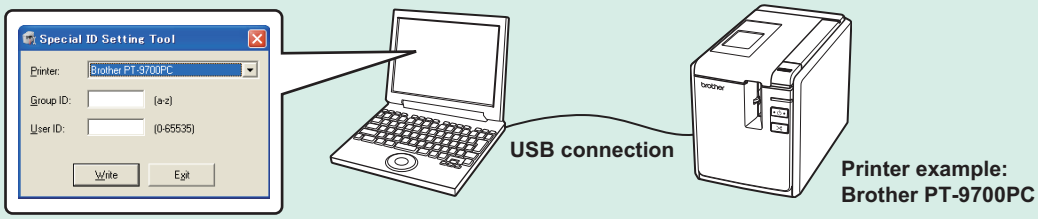
This document is written for use by system administrators and application developers.

Introduction

Thank you for purchasing a Brother printer.
The Special ID Setting Tool software allows you to assign a unique name (hereafter referred to as a “Special ID”) to a printer connected to the computer via USB cable.
A peripheral device can read the Special ID that was assigned to the printer using the Special ID Setting Tool, and you can create and implement programs that perform a process based on the ID that was read.
In this document, we will briefly explain how to use the Special ID Setting Tool and provide a sample source.
The Special ID Setting Tool can be used to help strengthen security and increase efficiency.

Stage 1: Assigning a Special ID

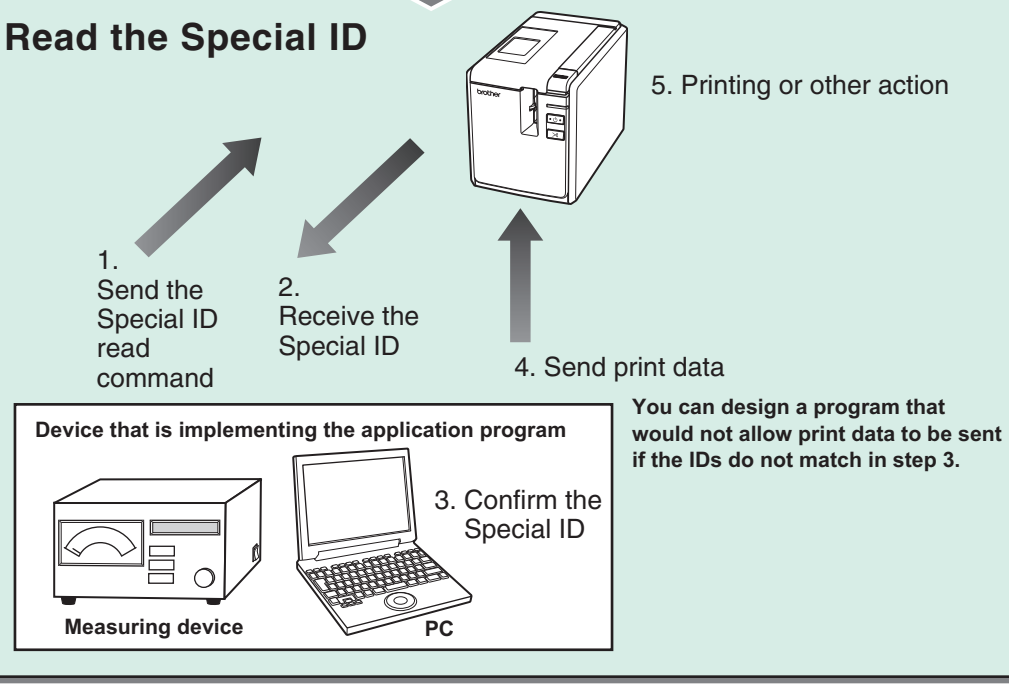
Use the Special ID Setting Tool to assign a Special ID to the printer.



Stage 2: Create the Application Program

Create a program to perform steps 1-5 of Stage 3 below, and implement the program with the peripheral device.

Stage 3: Read the Special ID



Trademarks

Main Specifications of the Special ID Setting Tool

Main Purpose

Assign a user-specified Group ID and User ID to the printer.

Supported Models (As of Jan, 2026)

- Brother PT-9700PC
- Brother PT-9800PCN
- Brother TD-4000
- Brother TD-4100N
- Brother PJ-622
- Brother PJ-623
- Brother PJ-662
- Brother PJ-663
- Brother QL-720NW
- Brother RJ-4030
- Brother RJ-4040
- Brother TD-2020
- Brother TD-2120N
- Brother TD-2130N
- Brother QL-800
- Brother QL-810W
- Brother QL-820NWB
- Brother PT-P900
- Brother PT-P900W
- Brother PT-P950NW
- Brother TD-4210D
- Brother TD-4410D
- Brother TD-4420DN
- Brother TD-4510D
- Brother TD-4520DN
- Brother TD-4550DNWB
- Brother TD-2030A
- Brother TD-2125N
- Brother TD-2125NWB
- Brother TD-2135N
- Brother TD-2135NWB
- Brother TD-2310D
- Brother TD-2320D
- Brother TD-2320DF
- Brother TD-2320DSA
- Brother TD-2350D
- Brother TD-2350DF
- Brother TD-2350DSA
- Brother TD-2350DFSA
- Brother TD-4420DNFC
- Brother TD-4550DNWBFC
- Brother TD-4215D
- Brother TD-4415D
- Brother TD-4425DN
- Brother TD-4525DN
- Brother TD-4455DNWB
- Brother TD-4555DNWB
- Brother TD-4425DNF
- Brother TD-4555DNWBF

Operating Environment

- Operating System
  Microsoft® Windows® 10
  Microsoft® Windows® 11

- Hardware
  USB port (USB 2.0 or higher)

Tool and Manual Download Site

You can download the latest version of the Special ID Setting Tool and this document from the following site:
https://www.brother.co.jp/eng/dev/index.aspx

Assigning a Special ID to the Printer

The below instruction illustrates with the case of PT-9700PC.

1 Install the Special ID Setting Tool to the desired directory on the PC.

2 Install the printer driver (corresponding to the printer to which a Special ID will be assigned) on the PC.

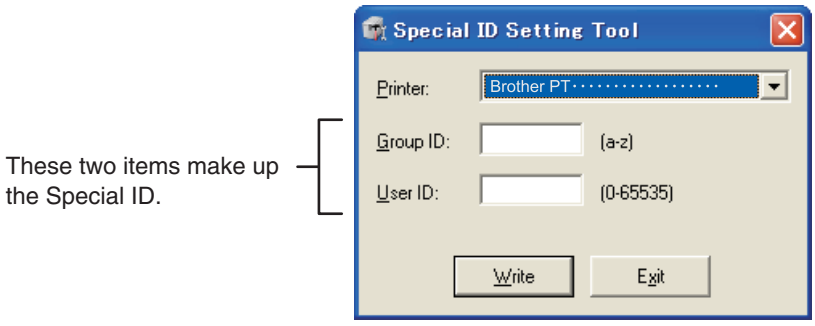
If the printer driver corresponding to your Brother machine is already installed, proceed to step 3.

3 Use a USB cable to connect the printer and the PC.

4 Turn the printer on.

Note: If the printer is not turned on, a Special ID cannot be assigned to that printer even if the printer is selected in step 6.

5 Start the tool by double-clicking "SpecialIDSettingTool.exe".



6 Select "Brother PT-9700PC" from the [Printer] drop-down menu.

7 Enter the desired Group ID in the [Group ID] field.

Up to four lower case letters (a-z) can be used.

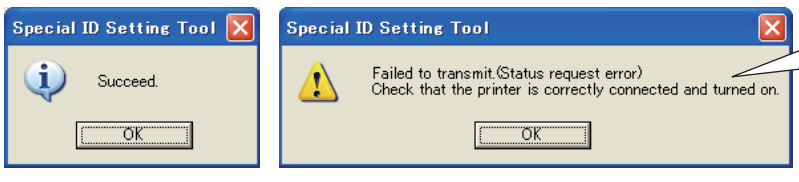
8 Enter the desired User ID in the [User ID] field.

- 0-65535 can be specified. If any value over 65535 is entered, 65535 becomes the specified value.
- You can only specify numbers; other characters cannot be entered.

9 Click [Write].

The Special ID is assigned to the printer. When the dialog box shown below on the left is displayed, proceed to step 10.
If the dialog box shown below on the right is displayed, click [OK] and resolve the problem according to the message that is displayed. Once the message is resolved, click [Write] again.

Note: Do not take the plug out of an electrical outlet or turn off the power button on the back of the machine immediately after clicking [Write]. The writing cannot complete correctly.



Write succeeded

Write error (example)

**An error such as this will be displayed if the printer is not turned on.
An error will also be displayed if the Group ID and/or User ID values are not specified.
In this case, a message prompting you to enter a value will be displayed.**

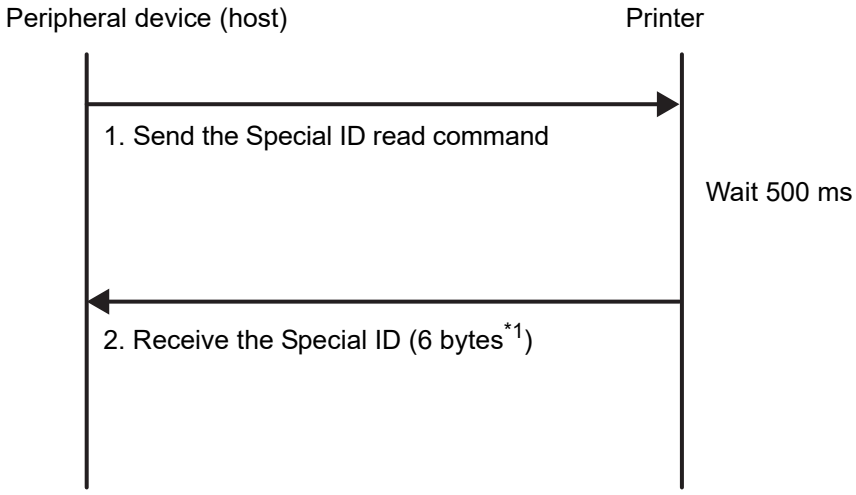
10 Click [OK].

11 Click [Exit] to finish.

Creating an Application Program

Data Communication Flow

The following describes the communication flow when the peripheral device (host) reads the printer's Special ID.



*1 Group ID (4 bytes) + User ID (2 bytes)

1 The peripheral device (host) sends bytes of data to the printer in order to read the Special ID.

The Special ID read command varies by printer model. Refer to the chart below when creating the application program and use the command corresponding to your printer.
The 500 ms wait time is only an approximation, but is the same for all models.

Model Name	Special ID Read Command
Brother PT-9700PC	{0x1B,0x69,0x61,0x01,0x55,0x08,0x7E,0x01,0x06,0x00}
Brother PT-9800PCN	
Brother PJ-622	
Brother PJ-623	
Brother PJ-662	
Brother PJ-663	
Brother TD-4000	{0x1B,0x69,0x61,0x01,0x55,0x08,0x63,0x00,0x06,0x00}
Brother TD-4100N	
Brother QL-720NW	{0x1B,0x69,0x61,0x01,0x55,0x08,0xF0,0x07,0x06,0x00}
Brother RJ-4030	
Brother RJ-4040	
Brother TD-2020	{0x1B,0x69,0x61,0x01,0x1B,0x69,0x55,0x49,0x01}
Brother TD-2120N	
Brother TD-2130N	
Brother QL-800	
Brother QL-810W	
Brother QL-820NWB	
Brother TD-4210D	
Brother TD-4410D	
Brother TD-4420DN	
Brother TD-4510D	

Brother TD-4520DN	
Brother TD-4550DNWB	
Brother TD-2030A	
Brother TD-2125N	
Brother TD-2125NWB	
Brother TD-2135N	
Brother TD-2135NWB	
Brother TD-2310D	
Brother TD-2320D	
Brother TD-2320DF	
Brother TD-2320DSA	
Brother TD-2350D	
Brother TD-2350DF	
Brother TD-2350DSA	
Brother TD-2350DFSA	
Brother TD-4420DNFC	
Brother TD-4550DNWBFC	
Brother TD-4215D	
Brother TD-4415D	
Brother TD-4425DN	
Brother TD-4525DN	
Brother TD-4455DNWB	
Brother TD-4555DNWB	
Brother TD-4425DNF	
Brother TD-4555DNWBF	
Brother PT-P900	{0x1B,0x69,0x61,0xFA,0x55,0x08,0xDC,0x01,0x06,0x00}
Brother PT-P900W	
Brother PT-P950NW	

Note: When the ID read command is sent, the printer's control command switches to raster mode. If you want to use ESC/P mode or P-Touch Template mode, refer to the chart below and add the command that switches the control command to your program so that it is sent after the ID read command.

Printer Control Command	Control Command Switching Command
ESC/P	{0x1B,0x69,0x61,0x00}
P-Touch Template	{0x1B,0x69,0x61,0x03}

2 The peripheral device (host) receives 6 bytes of data from the printer corresponding to the Special ID.

The following explains the structure of the Special ID. If the Special ID is noted in byte units, convert the specified Special ID to a hexidecimal value.

Example: If the Special ID Setting Tool was used to assign the Group ID:abcd and the User ID:1000.

	Group ID				User ID	
	a	b	c	d	1000	
	0x61	0x62	0x63	0x64	0xE8*2	0x03*2
Offset	0	1	2	3	4	5

*2 The User ID “1000” is displayed in Little Endian as “0xE8,0x03”.

Items Needed When Programming

■ USB Communication Library

In the provided sample source, the PTReadFile function and the PTWriteFile function are used as the printer communication functions. The following three files are needed in order to use these functions[*3]. The USB communication library is included with the Special ID Setting Tool.

- ptfilo.h
- ptfilo.dll
- ptfilo.lib

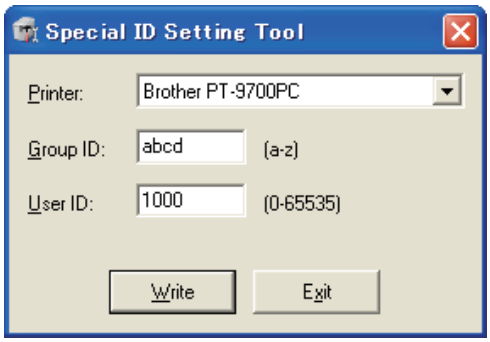
[*3] See “USB Communication Library Structure” (page 11) for more information.

■ Printer Driver Used When Setting the Special ID

Drivers can be downloaded from the Brother Solutions Center (http://solutions.brother.com/). Refer to the documentation included with the printer and install the driver correctly.
If the printer driver is already installed, there is no need to install it again.

REFERENCE: Sample Source Creation Environment and Special ID Settings

The following sample source uses the Windows® XP operating system and the code uses Visual Studio® 2005. Similar command sequences can be used for expressions when using other languages or other environments.
This sample source assumes that the Special ID Setting Tool has been used to assign a Special ID as shown on the right to a Brother PT-9700PC, and demonstrates a program that compares IDs and processes the compare result.



Sample Source

```
/*****************************************************/
//  Function: BOOL CompareID(void)
//  Purpose: Compare the ID read from the printer and the target ID
/*****************************************************/
BOOL CompareID(void)
{
  BYTE  pbyReadReq[]  = { 0x1B,0x69,0x61,          //Specifies the ID read command
                          0x01,0x55,0x08,          //(In this case the command for the PT-9700PC is specified.)
                          0x7E,0x01,0x06,0x00};
  BYTE  pbyReadID[6]  = {0};                       //This sequence stores the ID read from the printer
  DWORD dwResultSize  = 0;                         //This variable stores the number of bytes sent to the printer
  DWORD dwReadSize    = 0;                         //This variable stores the number of bytes actually read from the printer

  //
  //        Specify the target ID
  //
  BYTE  pbyTargetID[] = { 0x61,0x62,0x63,          //Specifies the comparison ID
                          0x64,0xE8,0x03};         //(In this case the target ID is "abcd1000".)

  //
  //        Send the ID read command to the printer
  //        (For details on the PTWriteFile function, see "Printer Communication Functions" (page 12).)
  //
  PTWriteFile( "Brother PT-9700PC",                //Specifies the printer name of the printer to be communicated with
               NULL,                               //Specifies the port name (e.g., "USB001") of the printer to be communicated with
                                                   //(In this case, the printer name has been specified, so the port name is NULL.)
               pbyReadReq,                         //Specifies the ID read command (10 bytes)
               sizeof(pbyReadReq),                 //Specifies the command size
               &dwResultSize,                      //Specifies the address where the number of bytes sent is stored
               5000                                //Specifies the timeout (5000 ms)
             );


  Sleep(500);                                      //Specifies the wait time (500 ms) between sending and receiving

  //
  //        Receive the ID sent from the printer
  //        (For details on the PTReadFile function, see "Printer Communication Functions" (page 12).)
  //
```

```
  PTReadFile(  “Brother PT-9700PC”,          //Specifies the name of the printer to be communicated with
               NULL,                         //Specifies the port name (e.g., “USB001”) of the printer to be communicated with
                                             //(In this case, the printer name has been specified, so the port name is NULL.)
               pbyReadID,                    //Specifies the buffer that receives the data
               sizeof(pbyReadID),            //Specifies the number of bytes to be received (6 bytes)
               &dwReadSize,                  //Specifies the address where the number of bytes in the read result is stored
               5000                          //Specifies the timeout (5000 ms)
             );

  //
  //        Compare the target ID and the read ID
  //
  if(memcmp(pbyTargetID,pbyReadID,6) == 0)
  {
        //When the IDs match:
        return TRUE;
  }else{
        //When the IDs do not match:
        return FALSE;
  }
}

//*************************************************************************/
//  Function: void Sample(void)
//  Purpose: Call the CompareID function, and branch the printing process (or other process) out based on the ID compare result
/*************************************************************************/
void Sample(void)
{
  //
  // Call the CompareID function, compare the IDs
  //
  if(CompareID())
  {
        /****This process is performed if the IDs match****/
        //(After sending the ID read command, the printer's control command switches to raster mode.)
        //(To use ESC/P mode or P-Touch Template mode, send the command noted in “Control Command Switching Command” (page 6).)
  }
}
```

The CompareID function reads the Special ID that was assigned to the printer using the Special ID Setting Tool and compares it to the target ID. The argument is void. TRUE is returned if the IDs match and FALSE is returned if the IDs do not match.
Here the Sample function determines whether or not the print process is performed, based on the result of the CompareID function.

USB Communication Library Structure

■ ptfilo.h (Header file)

Defines the PTWriteFile function and the PTReadFile function that are used for communicating with the printer. For details on these functions, see “Printer Communication Functions” (page 12).

■ ptfilo.dll (Dynamic Link Library file)

Store in a directory that the executable file can refer to.

■ ptfilo.lib (Library file)

An “Add Reference” is needed if you will use the library file. Specify “ptfilo.lib” as the name of the library to be used.
For information about “Add Reference”, we recommend referring to the Microsoft MSDN site. The following URLs are accurate as of December, 2010.

- http://msdn.microsoft.com/library/hcce369f(v=VS.80).aspx
- Search for “.lib Files as Linker Input” at http://msdn.microsoft.com/en-us/default.aspx

■ BOOL PTWriteFile (pPrinterName, pPortName, pBuffer, dwWriteSize, pResultSize, dwTimeout)

```
LPSTR       pPrinterName;   //The name of the printer to be written to
LPSTR       pPortName;      //The name of the port to be written to
LPBYTE      pBuffer;        //The address of the data written to the printer or port
DWORD       dwWriteSize;    //The number of bytes written
LPDWORD     pResultSize;    //The address of the number of bytes written
DWORD       dwTimeout;      //Timeout
```

The PTWriteFile function writes data to the printer or port. If the specified printer or port is in use, this function tries writing again until it times out.

Parameter	Description
pPrinterName	A pointer that identifies the printer to be written to by specifying characters that end in NULL. If this parameter is NULL or refers to an invalid printer, the function uses the pPortName parameter. If this parameter is valid, the port used by the specified printer is used as the output destination.
pPortName	A pointer that determines the port to be written to by specifying characters that end in NULL. If the pPrinterName parameter is valid, this parameter is disregarded. If the name of the port used by the printer is known, specify the port name for faster processing.
pBuffer	A pointer that identifies the buffer containing the data written to the printer or port.
dwWriteSize	Specifies the number of bytes written to the printer or port.
pResultSize	Specifies the number of bytes written when this function is called. The PTWriteFile function sets this value to 0 before performing error checks or other operations.
dwTimeout	Specifies the timeout (in ms) of the write operation. Control is returned if the function times out, even if it is still performing the write operation. If 0 is specified for dwTimeout, control is returned as soon as one write is performed. In this case, the function is limited so that up to 64 bytes can be written. If INFINITE is specified for dwTimeout, the timeout feature does not operate for the function.

Returned Value

If the function ends normally, a TRUE is returned. Otherwise a FALSE is returned. Note that even if the function is successful, this does not necessarily mean that all data has been written. Confirm that the specified number of bytes (dwWriteSize) and the actual number of bytes written (pResultSize) are the same. Additionally, use the GetLastError function to retrieve extended error information.

■ BOOL PTReadFile (pPrinterName, pPortName, pBuffer, dwReadSize, pResultSize, dwTimeout)

```
LPSTR       pPrinterName;   //The name of the printer to be read from
LPSTR       pPortName;      //The name of the port to be read from
LPBYTE      pBuffer;        //The address of the buffer that receives the data
DWORD       dwReadSize;     //The number of bytes read
LPDWORD     pResultSize;    //The address of the number of bytes read
DWORD       dwTimeout;      //Timeout
```

The PTReadFile function reads data from the printer or port. If the specified printer or port is in use, this function tries reading again until it times out.

Parameter	Description
pPrinterName	A pointer that identifies the printer to be read from by specifying characters that end in NULL. If this parameter is NULL or refers to an invalid printer, the function uses the pPortName parameter. If this parameter is valid, the port used by the specified printer is used as the read input destination.
pPortName	A pointer that identifies the port to be read from by specifying characters that end in NULL. If the pPrinterName parameter is valid, this parameter is disregarded. If the name of the port used by the printer is known, specify the port name for faster processing.
pBuffer	A pointer that identifies the buffer receiving the data read from the printer or port.
dwReadSize	Specifies the number of bytes read from the printer or port.
pResultSize	Specifies the number of bytes read when this function is called. The PTReadFile function sets this value to 0 before performing error checks or other operations.
dwTimeout	Specifies the timeout (in ms) of the read operation. Control is returned if the function times out, even if it is still performing the read operation. If 0 is specified for dwTimeout, control is returned as soon as one read is performed. In this case, the function is limited so that up to 64 bytes can be read. If INFINITE is specified for dwTimeout, the timeout feature does not operate for the function.

Returned Value

If the function ends normally, a TRUE is returned. Otherwise a FALSE is returned. Note that even if the function is successful, this does not necessarily mean that all data has been read. Confirm that the specified number of bytes (dwReadSize) and the actual number of bytes read (pResultSize) are the same. Additionally, use the GetLastError function to retrieve extended error information.