EPSON
EXCEED YOUR VISION

Printing Using Multiple Interfaces Manual

Printing Using Multiple Interfaces

Version 1.14.6 Dec. 2017

Notes

(1) Reproduction of any part of this documentation by any means is prohibited.
(2) The contents of this documentation are subject to change without notice.
(3) Comments and notification of any mistakes in this documentation are gratefully accepted.
(4) This software cannot be used with other equipment that the specified.
(5) EPSON will not be responsible for any consequences resulting from the use of any information in this documentation.

Contents

Chapter 1 Preface

This Manual is an application development guide for conducting Printing Using Multiple Interfaces. Read this Manual before you begin programming and reference the information contained in this Manual as needed.

1.1 Characteristics

You can select two interfaces from among the various interfaces installed on a single printer to print from separate applications.

In this Manual, application refers to primary system applications and subsystem applications.

- A primary system application is assumed to be an existing application.
- A subsystem application is assumed to be a new application. This refers to an application added to existing applications.

  For example, applications used for printing to a kitchen printer or from a tablet.

Unless otherwise stated, this Manual describes development methods for subsystem applications.

Chapter 2 Notes of caution

- Exclusive access rights via the Claim method should be acquired and released for each printing. While exclusive access rights are claimed, printing is not allowed from primary system applications.
- Do not print long receipts. Less than one second per print is recommended.
  Printing lasting longer than one second can result in an error when printing from a primary system application.
- In the event of an error, immediately release the exclusive access rights.
- Only supported on printers.
- SetBitmap method is not supported.
- Use the Claim method to initialize the device. If the device produces an error, the Claim method will return an error.
- For primary system applications, do not use the Marque function in the UPOS LineDisplay.
  You cannot print from a subsystem application while the LineDisplay Marque function is running on a primary system application.
  To run the Marque function, use the LineDisplay DirectIO method.
  Refer to "3.2.2 Marque Programming" for user instructions.
- LineDisplay is delayed during printing from a subsystem application, so wait until printing is completed to display the information.

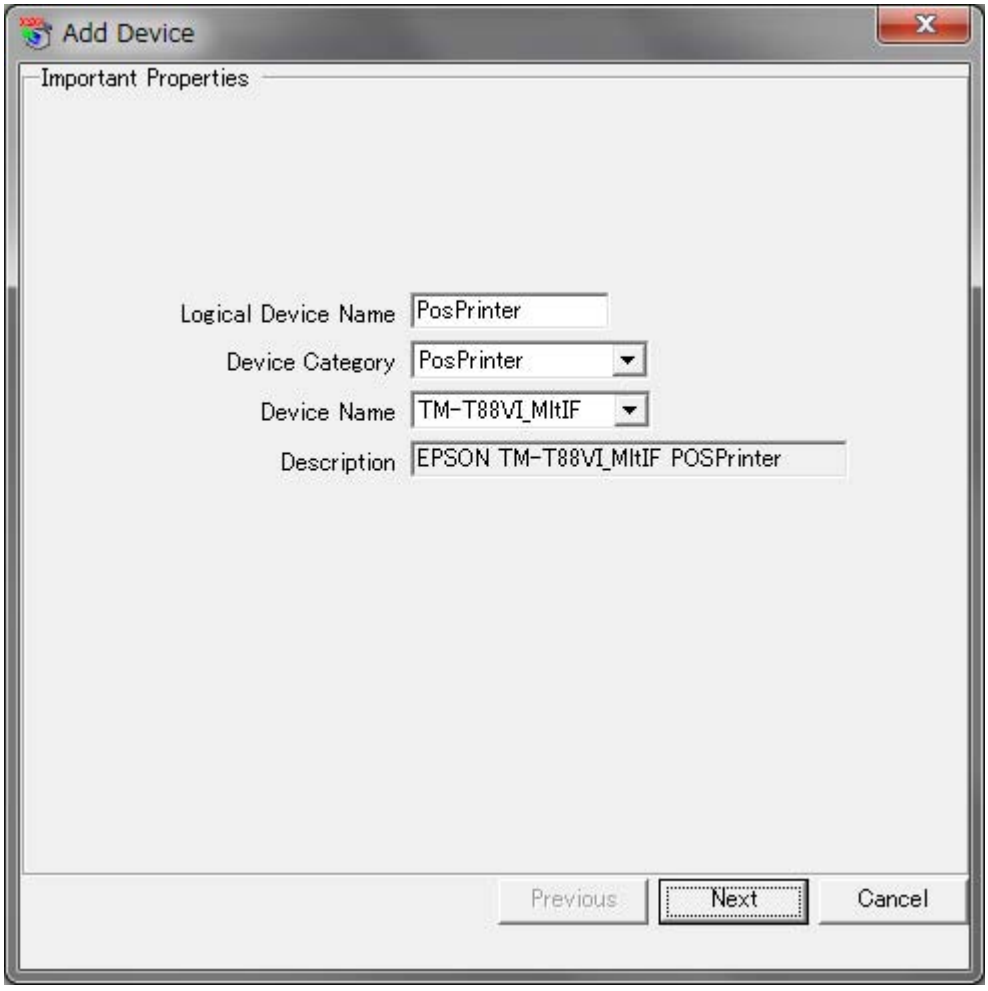
Chapter 3 User instructions

This section explains device registration and programming methods.

3.1 Device registration

To register a device using the SetupPOS utility, make sure to select a device with "_MltiIF" for a device name suffix.

- Registration for subsystem applications



3.2 Programming method

3.2.1 Print programming

The process for creating a subsystem application is as follows.

- Driver initialization process: Run open process
- Print process: Acquire and release exclusive access rights, run print
- Driver end process: Run close process

Sample code is indicated below.

```
DeviceInfo deviceInfo = posExplorer.GetDevice(DeviceType.PosPrinter,strLogicalName);
PosPrinter ptr = (PosPrinter)posExplorer.CreateInstance(deviceInfo);

void open(String LogicalDeviceName){
        try{
                ptr.Open();
        }
        catch(PosControlException){}
}
void print(){
        try{
           ptr.Claim(1000);
           ptr.DeviceEnabled = true;
           ptr.PrintNormal(PrinterStation.Receipt, "PrintData" + "¥n");
        }
        catch(PosControlException){}
        finally{
                try {
                        ptr.DeviceEnabled = false;
                        ptr.Release();
                }
                catch (PosControlException) {}
        }
}
void close(){
        try{
                ptr.Close();
        }
        catch(PosControlException){}
}
```

3.2.2 Marque program

Use the following sample code to run a LineDisplay Marque on a primary system application.

[Marque start]

```
deviceInfo = posExplorer.GetDevice(DeviceType.LineDisplay,strLogicalName);
LineDisplay disp = (LineDisplay)posExplorer.CreateInstance(deviceInfo);
int pram2 = 0;
byte[] pram3 = {
        0x1B, 0x3D, 0x02, //ESC = (specifies command sent to LineDisplay)
        0x1F, 0x03, // US MD3 (specifies horizontal scroll)
        0x1F, 0x3A, // US: (Starts macro definition)
        0x0C, // CLR (Clears display screen)
        0x1F, 0x0D, // US CR (Moves display position to right edge of current row)
        0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, //display data
        0x1F, 0x3A,    // US: (macro definition end)
        0x1F, 0x5E, 0x50, 0x20 // US^ n m (run macro)
        // n sets the character display interval during macro process run to
          n x 20 msec
        // m sets the idle time interval during macro process repeat to m x 50 msec
              };
disp.DirectIO(EpsonLineDisplayConst.DISP_DI_OUTPUT, pram2, pram3);
```

[Marque stop]

Marque stop programming not required. The Marque will stop by running the LineDisplay process during the Marque.

Chapter 4 Error codes

Error code has been changed by the following method.

Method name	ResultCode	ErrorCodeExtended	Meaning
Claim	E_ILLEGAL	EX_FAIL_MULTI_INTERFACE_PRINTING	Printing or printer error status.
		EX_INVALID_VALUE	Parameter value not defined by UPOS has been set.
		EX_BADPORT	Invalid device connected.
		0	Cannot connect to network.
	E_CLOSED	0	Service is not open.
	E_TIMEOUT	0	Could not acquire exclusive access rights during specified time.
	E_OFFLINE	0	Power recovery process running.
		EX_BADPORT	Communications port is being used by another device.
SetBitmap	E_ILLEGAL	EX_NOTSUPPORTED	Not supported.