

Printing Using Multiple Interfaces Manual

Printing Using Multiple Interfaces

Version 3.00 Feb. 2019

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 than 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.....	1
	CHARACTERISTICS	1
CHAPTER 2	NOTES OF CAUTION.....	2
CHAPTER 3	USER INSTRUCTIONS	3
3.1	DEVICE REGISTRATION	3
3.2	PROGRAMMING METHOD.....	4
3.2.1	<i>Print programming</i>	4
3.2.2	<i>Marque program</i>	5
CHAPTER 4	ERROR CODES	6

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.

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 "_MltIF" for a device name suffix.

■ Registration for subsystem applications

Add New Device [Select DeviceNameKey]

Add New Device

Select Device Name: TM-m30_MltIF

Select detailed model: TM-m30_MltIF

☐ Display Ver1xx ☒ Display Ver2xx

Device Description: EPSON TM-m30_MltIF POS Printer

INF File Name: C:\PROGRA~2\OPOS\Epson2\Ppm30MltIF.INF Browse...

Add New LDN

Set up a logical device name if necessary.

* A logical device name isn't indispensable.
A logical device name isn't set up in the case as the blank.
And, a logical device name can be set up even later.

< Back Next > Cancel Help

3.2 Programming method

3.2.1 Print programming

Please refer "Printer "of the VC++ sample program when you implement subsystem application.

- 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.

```
void Open(LPCTSTR LogicalDeviceName){
    m_Ptr1.Open( LogicalDeviceName );
}
void print(){
    m_Ptr1.ClaimDevice( 1000 );
    m_Ptr1.SetDeviceEnabled( TRUE );

    m_Ptr1.PrintNormal( PTR_S_RECEIPT, "PrintData¥n" );

    m_Ptr1.SetDeviceEnabled( FALSE );
    m_Ptr1.ReleaseDevice();
}
void close(){
    m_Ptr1.Close();
}
```

3.2.2 Marque program

Please refer "Display" of the VC++ sample program when you implement the marquee function for LineDisplay in primary system application.

[Marque start]

```
long pData;
```

```
CString cString;
```

```
BSTR pString;
```

```
pData = 0;
```

```
cString = CString(
```

```
    _T("\x1b\x3d\x02 // ESC = (specifies command sent to LineDisplay)
```

```
    \x1f\x03 // US MD3(specifies horizontal scroll)
```

```
    \x1f\x3a // US:(Starts macro definition)
```

```
    \x0c // CLR(Clears display screen)
```

```
    \x1f\x0d // US CR(Moves display position to right edge of current row)
```

```
    \x30\x31\x32\x33\x34\x35\x36\x37\x38\x39 // display data
```

```
    \x1f\x3a // US:(macro definition end)
```

```
    \x1f\x5e\x50\x20") // 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
```

```
, 26 ); //The number of bytes of the transmission command.
```

```
pString = cString.AllocSysString();
```

```
m_Displ.DirectIO( DISP_DI_OUTPUT_NORMAL, &pData, &pString );
```

```
::SysFreeString( pString );
```

[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	ResultCodeExtended	Meaning
Claim	E_ILLEGAL	OPOS_EX_FAIL_MULTI_INTER FACE_PRINTING	Printing or printer error status.
		OPOS_EX_INVALID_VALUE	Parameter value not defined by UPOS has been set.
		OPOS_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
		OPOS_EX_BADPORT	Communications port is being used by another device.
SetBitmap	E_ILLEGAL	OPOS_EX_NOTSUPPORTED	Not supported.