TOSHIBA

TOSHIBA TEC SINGAPORE PTE. LTD.

Specification For : Toshiba Windows Driver Manual

(Software / ~~Product~~)

Model : TRST-P1X, TRST-P2X, TOSHIBATEC KOP-3X, TOSHIBATEC KOP-3S06 and TRST-L1X

25th Revision : (15-November-2024)

CHANGE RECORD				
REV	DATE	Page	DESCRIPTION OF CHANGE	APPROVAL
01	20/Apr/20		Initial version	
02	24/Apr/20	P4 P13 P22-23 P28-30 P36 P39-41 P43 P43 P58-66	Added information in “Overview” Added information for driver location Modify instruction on “Add Printer with LAN Interface” Added information to change the Printer Interface Update information on uninstallation process Update example of True Type Font Substitution Update information for default log file location Added explanation about ControlA Font Added Printer Device Font	
03	08/May/20	P49-50 P56-57 P63	Modify Document Layout Modify Document Layout Remove comment box that not refer anything	
04	22/May/20	P1,7,61,62 P4,7,13,31 P29-30 P42-43 P64-68	Change TRSTPXX ==> TRST-P1X and TRST-P2X Driver name change to TEC WinDriver Additional information about setting Serial communication parameter Added Example for Control & ControlA Font Remove Italic & Bold Font from sample code	
05	15/June/20	P4 P6,10 P7	Change versioning from V3.XX to V1.3.XX.XXX Change GUI & details for reboot after installation Update folder installation GUI	
06	03/Aug/21	Pi P5,P9	Remove confidential and remove approved table Remove GUI for selected Printer Model	
07	04/Aug/21	P4 P52 P49-P59	Change versioning from V1.3.XX.XXX to V3.21.X.X Remove Watermark Section Change GUI after removing Watermark tab in Printing Preferences	
08	29/Dec/21	P4	Add windows 11 (64 bit) as supported OS Change versioning from V3.21.X.X to V3.22.X.X	
11	20/Aug/22	P59-P60	Add QR Code in 2D Code Tab	
12	11/Aug/22	P4 P10	Add windows Server 2016 / 2019 as supported OS Added note for Windows Server 2016 / 2019 to use manual install	
13	23/Sep/22	P49-P52 P70	Added Print Option -> Shrink to Fit on Layout Tab Added Restrictions and Cautions	

14	27/Dec/22	i P4 P5 P7 P21 P28 P43 P46 P61 P62 P63	Add TOSHIBATEC KOP-3X model Change for specification product to Toshiba Windows Driver Manual Add GS1 DataBar into table of content Added KOP-3X in list of installed printer Change installer version from V3.23.X.X to Vx.x.x.x Add note need to reboot pc after installation complete Add note KOP-3X printer not support serial interface Add note KOP-3X printer not support LAN interface Add note KOP-3X printer not support cash drawer Add location Log File Path for KOP-3X printer Add GS1 DataBar 2DCode Add note KOP-3X printer not support GS1 DataBar Change version image of printer properties UI Change version image of printing preferences UI	
15	20/Feb/23	P51	Change radio button to label text of printer resolution	
16	17/Mar/23	P74	Add limitation for pause print job in print queue	
17	12/June/23	i P5 P7 P8 P37 P46 P50,P64 P51 P52,P66 P53 P32,P39, P76-P77	Add TOSHIBATEC KOP-3S06 model Add list for supported OS Remove installer version V10.0.1.0 Change image of icon printer Change image of installer package Change printer list to list table Add location log for KOP3S06 Change version tab image Add note for version tab Change image of main tab Add partial cut definition Add note for partial cut Change image of version tab Change paper list to list table Add silent installation Add silent uninstallation	
18	31/Aug/23	P79-93	Added API detail	
19	04/Dec/23	P57	Change picture of document settings tab	

20	13/Dec/23	i P7 P10 P40 P50 P56 P58 P68 P81	Add TRST-P3X Printer model Added TRST-P3X in list of installed printer Change picture due to add TRSTP3X printer Add TRSTP3XLMN.dll to table Add location of TRST-P3X directory log Add paper support in rose printer Update document settings tab UI Add option buzzer description Add note for printers support option buzzer Add DataMatrix UI and description Add TRST-P3X API support	
21	15/Mar/24	P57 P93-P96 P102-P103	Add Auto paper size definition Add new support buzzer API Add Auto paper size check box restrictions and cautions	
22	22/June/24	P55 P60-P61 P69 P71 P72-P73 P100	Add note module width support for TRST-P3X is from 2-8 Add Aztec Code 2D Code Add MaxiCode 2D Code Add TPPIGetInterfaceType function to get driver interface Remove partial cut in main tab Add partial cut in paper source setting Change custom paper source setting method	
23	15/July/24	Cover, P7, P10, P40, P50, P54, P56-P57, P61-P62, P70-P72, P74-P76, P87, P102, P108, P109-P111	Change TRST-P3X to TRST-L1X Add limitation for RS232 and LAN interface related to paper removal sensor	
24	20/Sep/24	P61	Remove partial cut section Add information about cut support for each printers	
25	16/Dec/24	P7 P43 P57 P87 P109	Remove support on windows 7 Added KOP-3S01 and KOP-3S01-A printer model Added picture comment for TRST-P2N2-XX printer model number Added 82mm paper size support for KOP-3X printer Remove support on windows 7 and added support for windows 11 Changing printer model from KOP-3S01 to KOP-3S01-A or vice versa	

Table of Contents

1. Overview

This document contains information about Windows Printer Driver installation, Printer Properties and Printing preferences provided for the following printers:

- TRST-P1X / HSP-150
- TRST-P2X / HSP-100
    - TRST-P2N-XX (Support partial cut)
    - TRST-P2N2-XX (Support partial cut and full cut)
- TOSHIBATEC KOP-3X
    - KOP-3S01 (Support 58 mm and 80 mm paper size)
    - KOP-3S01-A (Support 58 mm, 80 mm and 82 mm paper size)
- TOSHIBATEC KOP-3S06
- TRST-L1X

These instruction supported for following operating system:

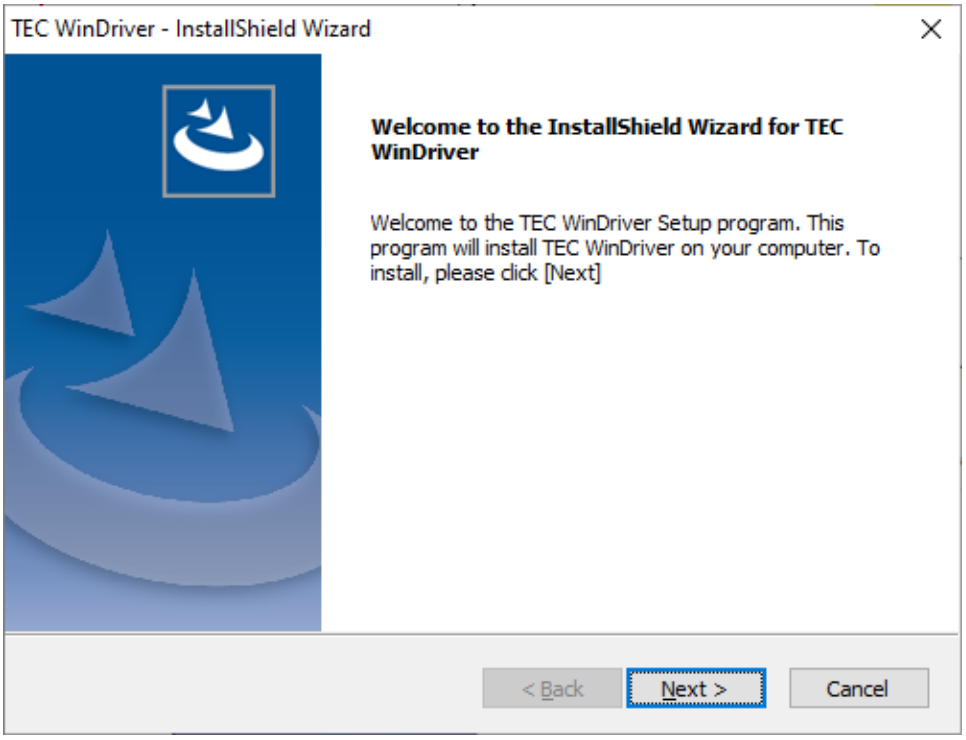
- Windows 10 (32 bit & 64 bit)
- Windows 11 (64 bit)
- Windows Server 2016
- Windows Server 2019

2. How to install the Printer Driver

2.1 Install by using installer application (USB connection)

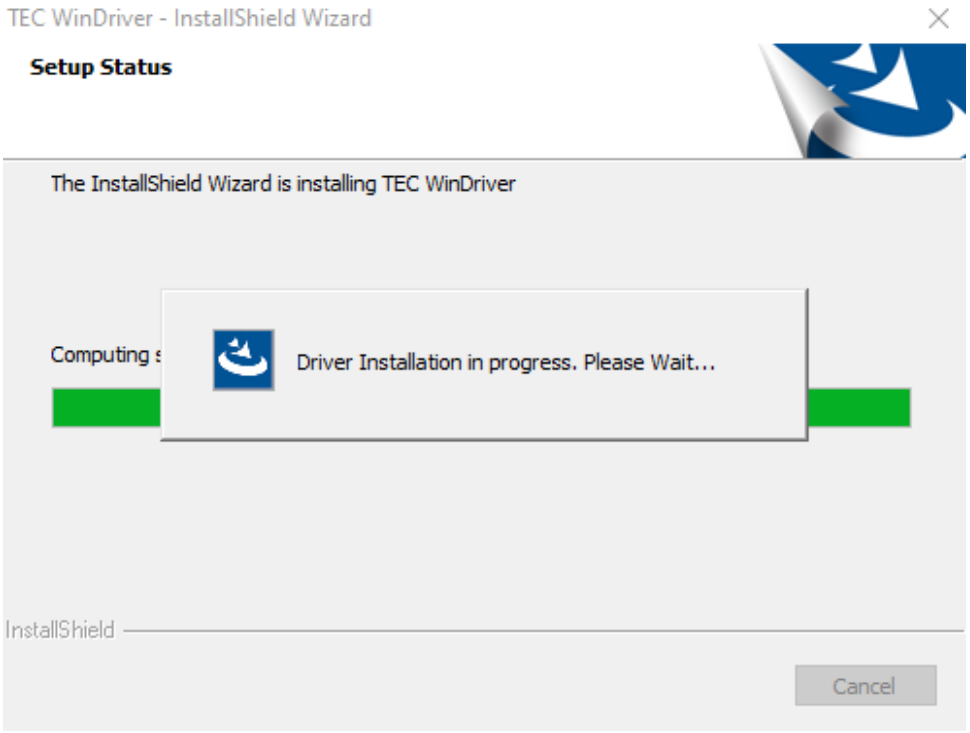
To start the installation, please select and run the “TEC_WinDriver_Vx.x.x.x”

Below is Welcome Dialog after run the installer. Click "Next >".

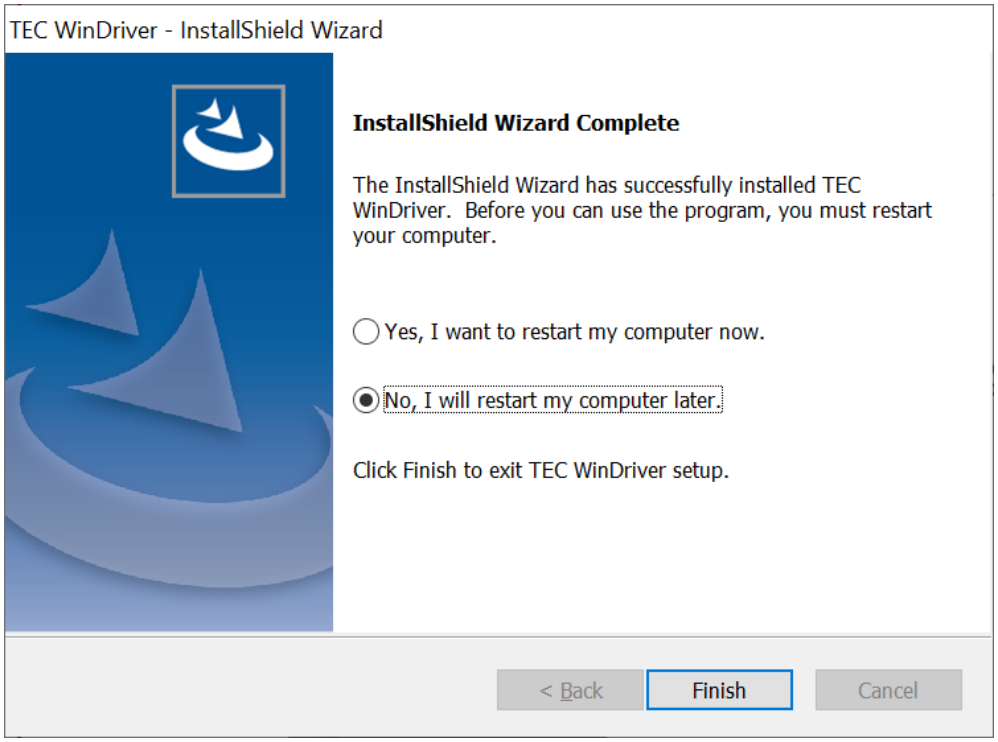


Select Accept the license agreement option. Click "Next >".

Driver Installation Process. Please wait until installation finish



Installation complete. In some case during updating the driver, the printer properties and printer preferences version still refer to previous driver so it is recommended to restart the PC when you perform update driver.

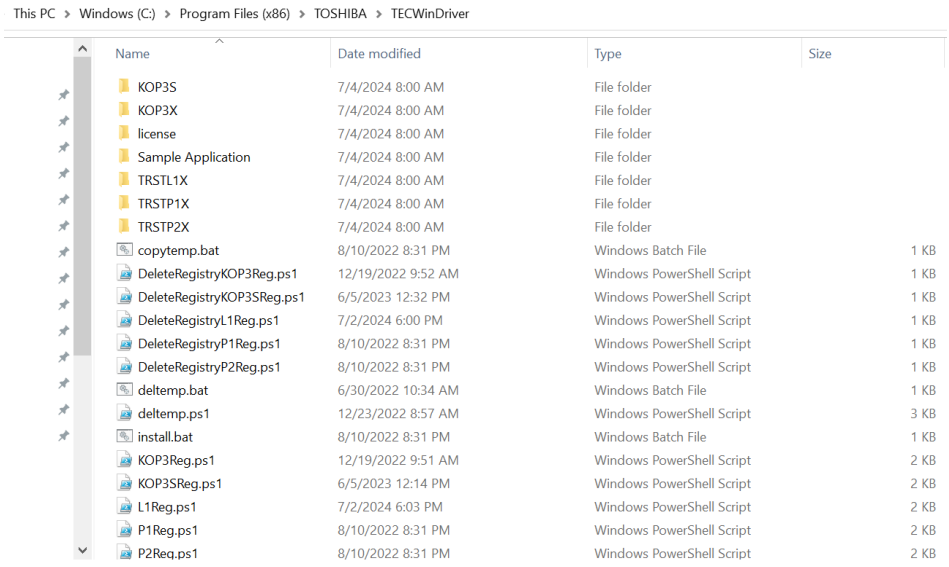


Click "Finish" to complete the installation.

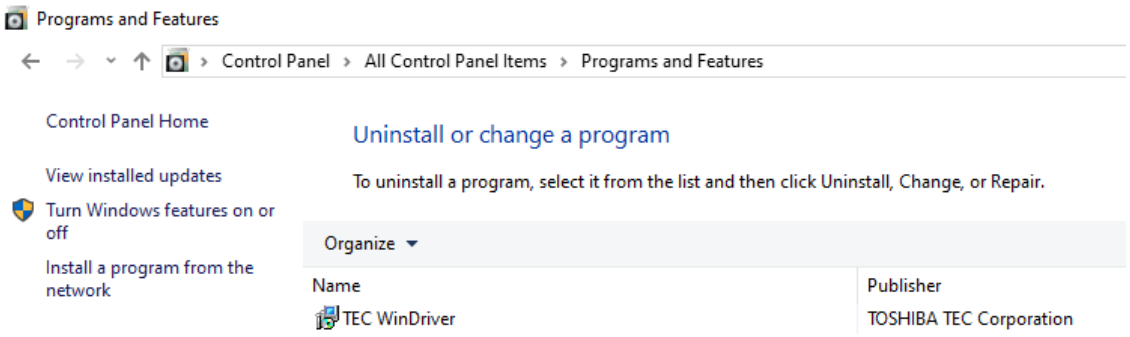
After installation finished, you will see the printer model as below icon in the "Devices and Printers" folder.

Note:
If old version of windows driver already installed in the pc, please restart the computer first before use the new windows driver version. **“XXXXXX”** is printer name, you can refer in the overview.

Driver Package Location → C:\Program Files (x86)\TOSHIBA\TECWinDriver

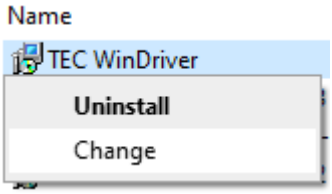


The Installer program call “TEC WinDriver” will appear on “Programs and Features”

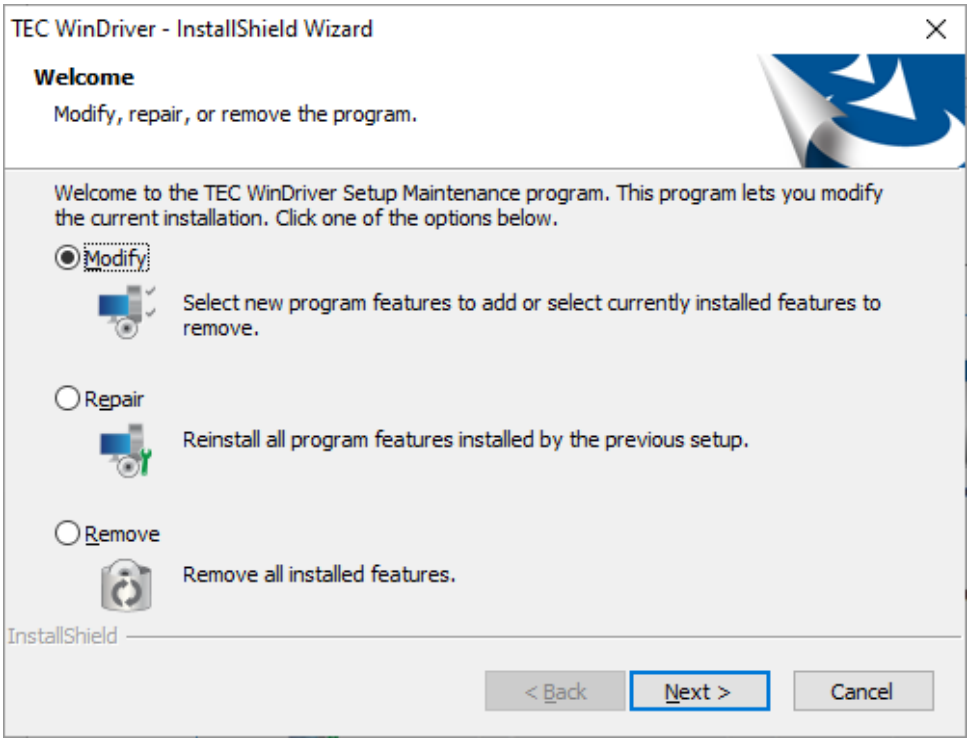


Note:

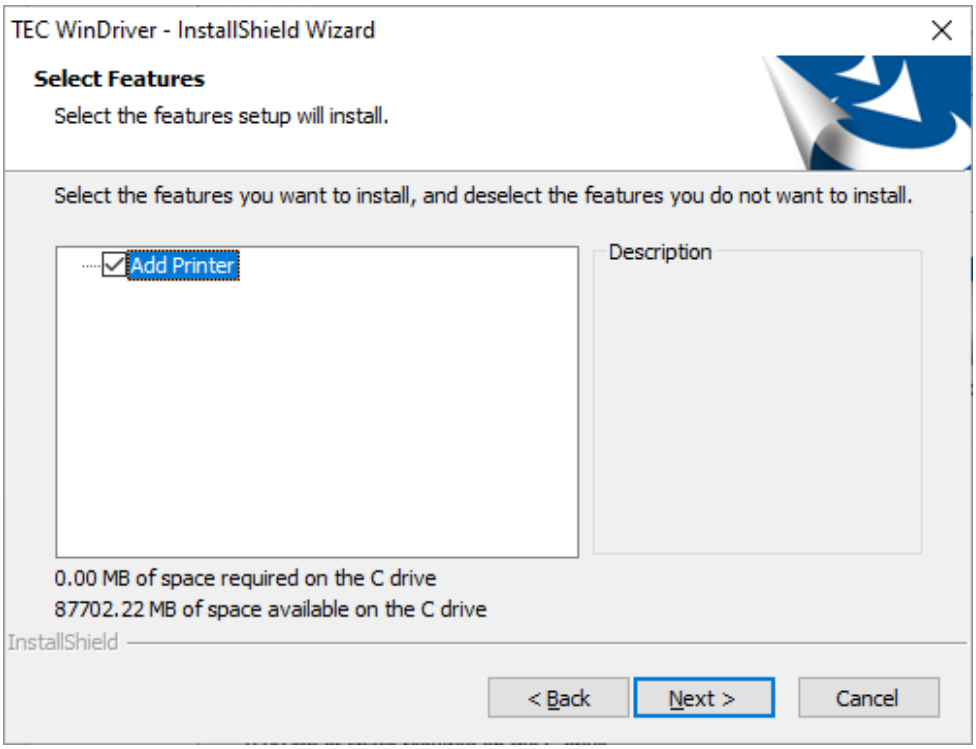
To add another printer model, please right click the “TEC WinDriver” program on “Program and Features” then select “Change”



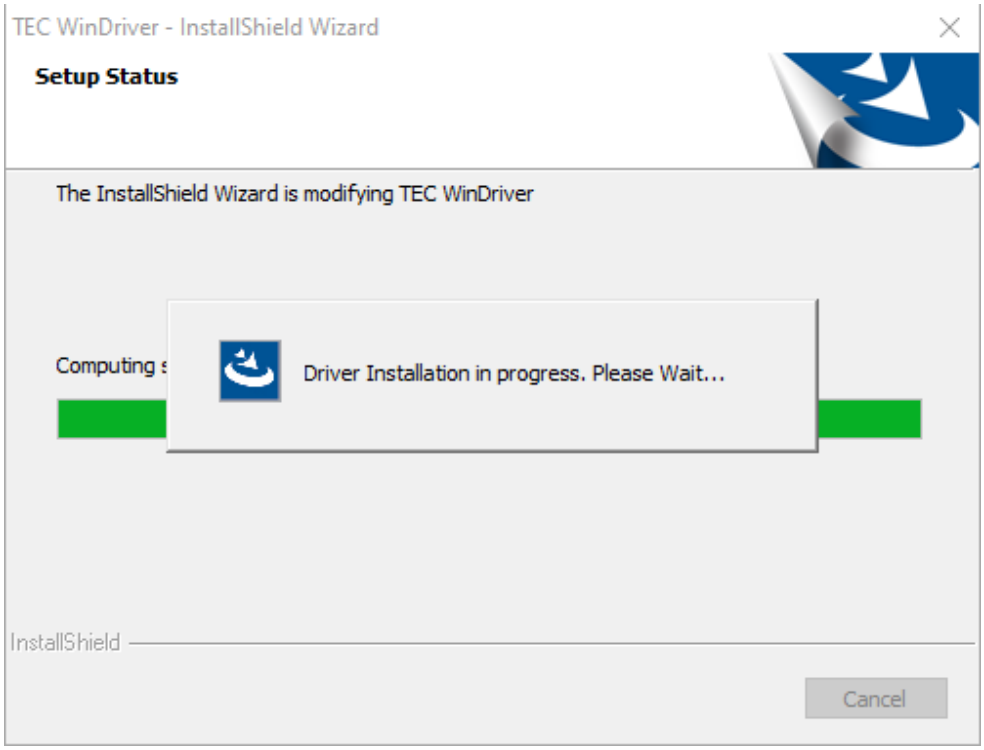
Select Modify option then click "Next >".



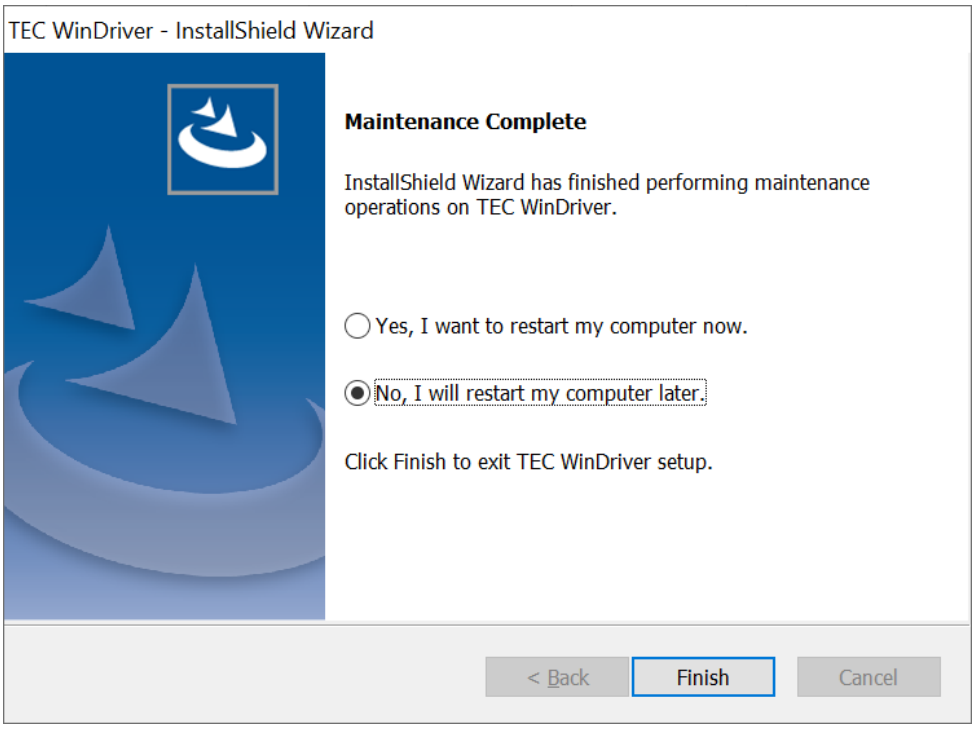
Select Add Printer then click "Next >".



Driver Installation Process. Please wait until installation finish



Installation complete. In some case during updating the driver, the printer properties and printer preferences version still refer to previous driver so it is recommended to restart the PC when you perform update driver.



Click "Finish" to complete the installation.

2.2 Manual install by using “Add Printer” Wizard

Note: For Windows Server 2016 / 2019 is recommended use this method

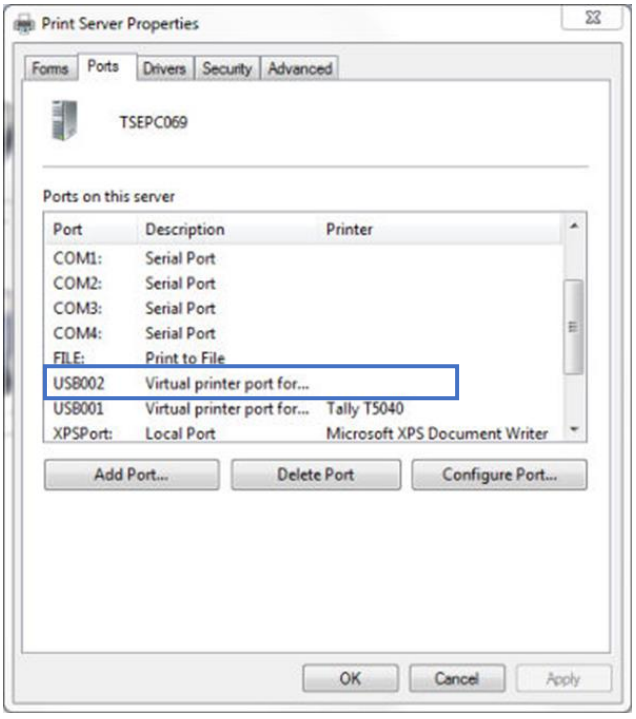
2.2.1 Add Printer with USB Interface

Below is step to install the driver by using USB interface

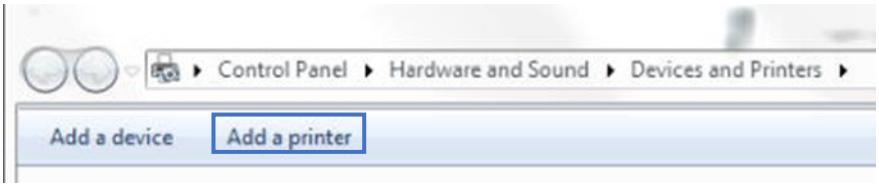
1) Connect printer to PC via USB cable.

It will appear in “Devices and Printers” under Unspecified like below.

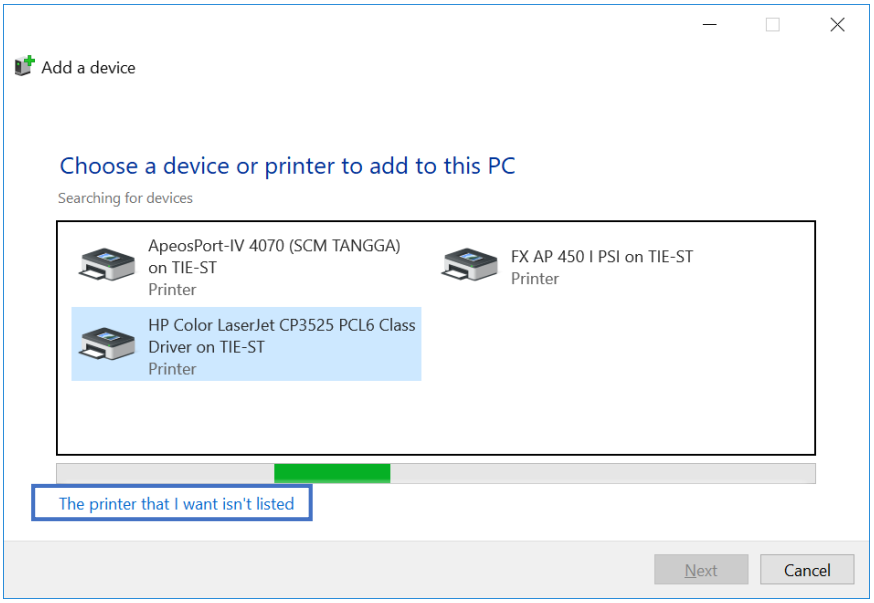
It will create new USB port like below



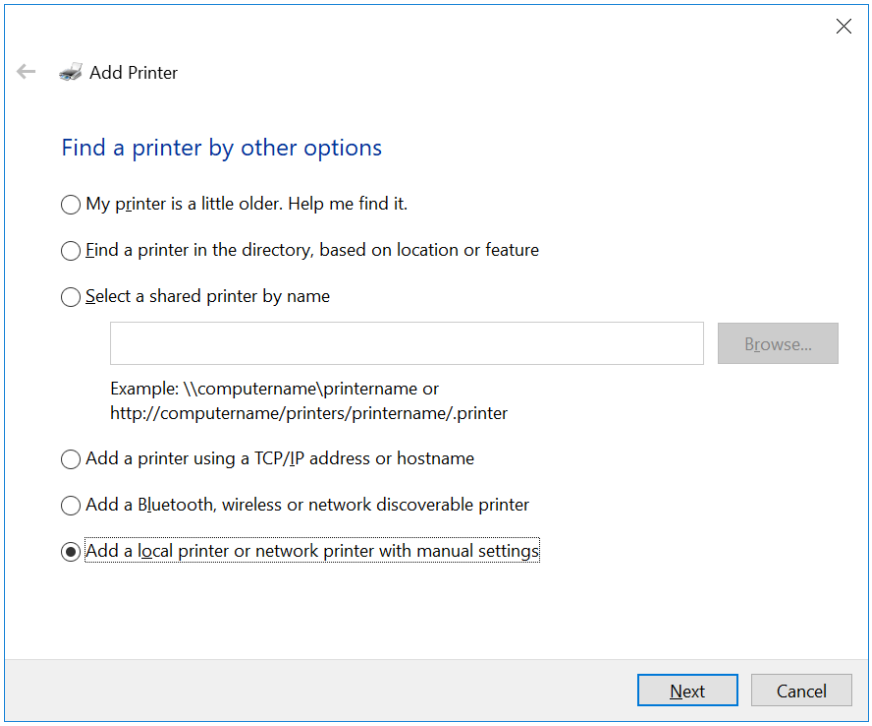
2) Click [Add a printer]



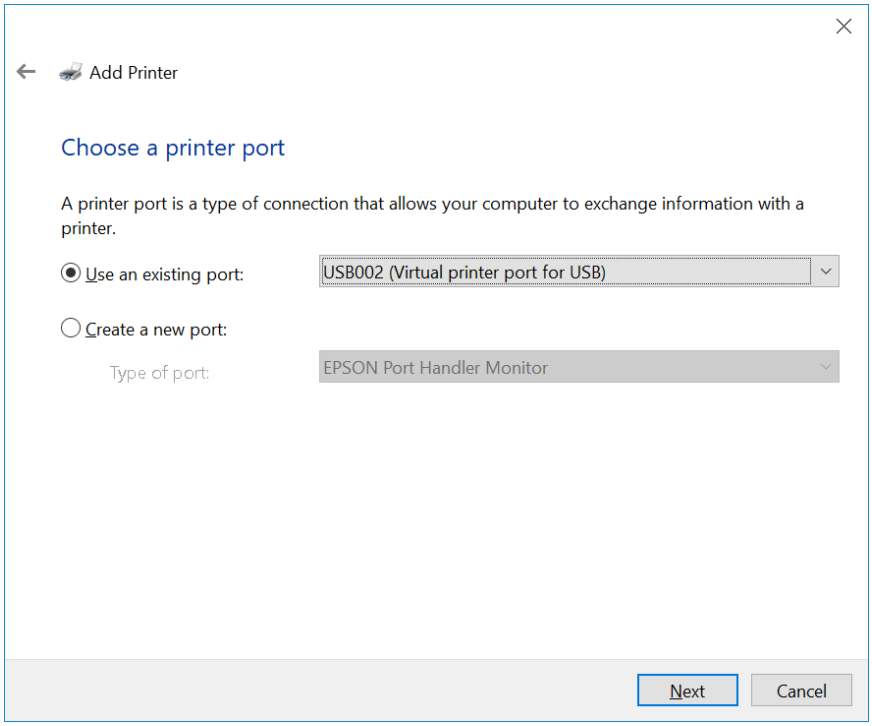
3) Click “The printer that I want isn’t listed”



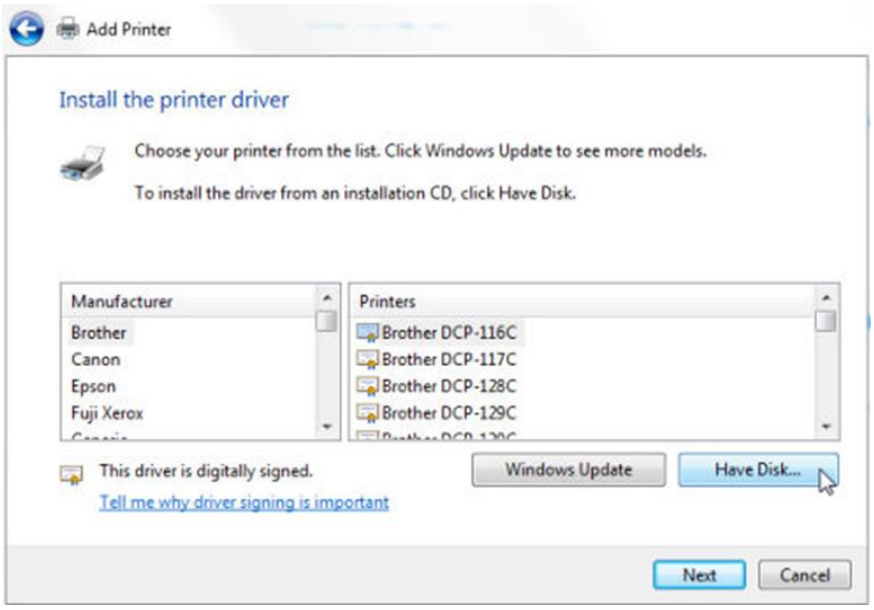
4) Choose “Add a local printer or network printer with manual settings” and click [Next].



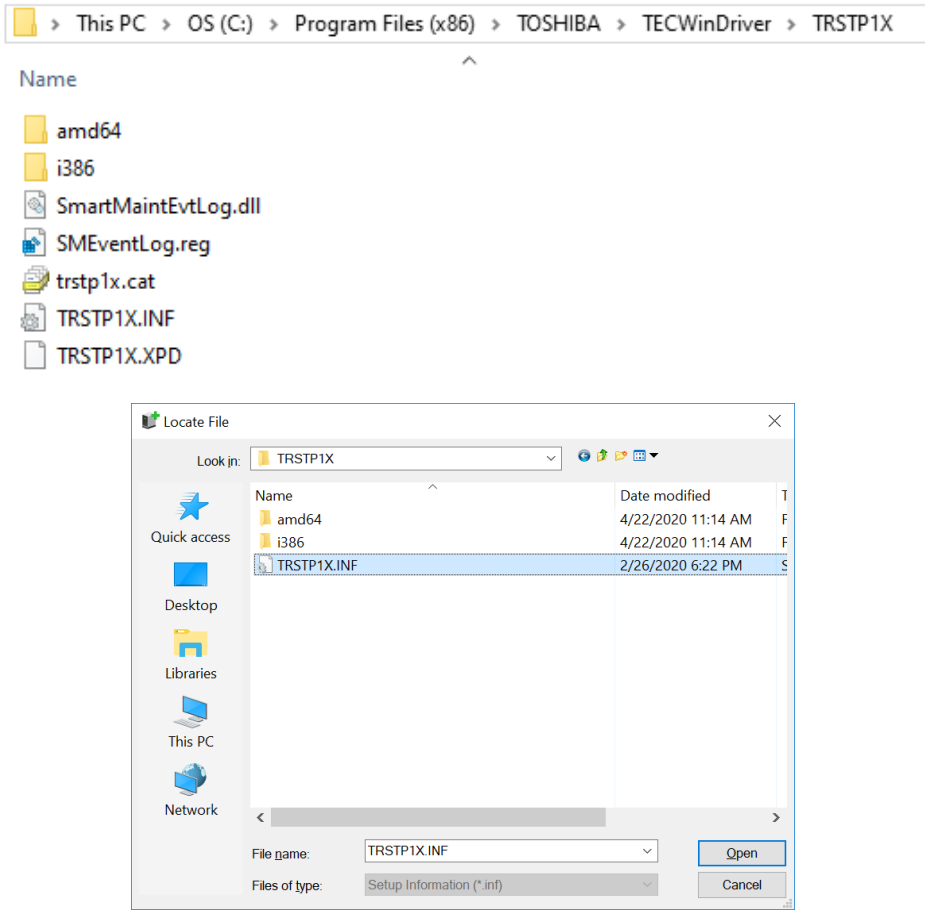
5) Choose a printer port. Select the newly created USB port in Step #1 and click [Next].



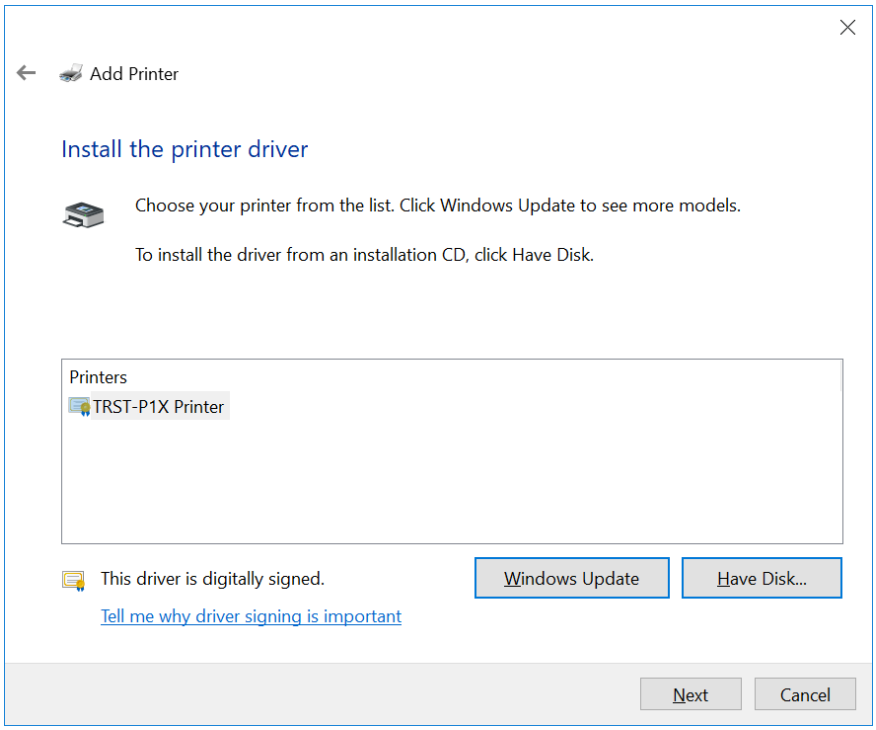
6) Click [Have Disk…]

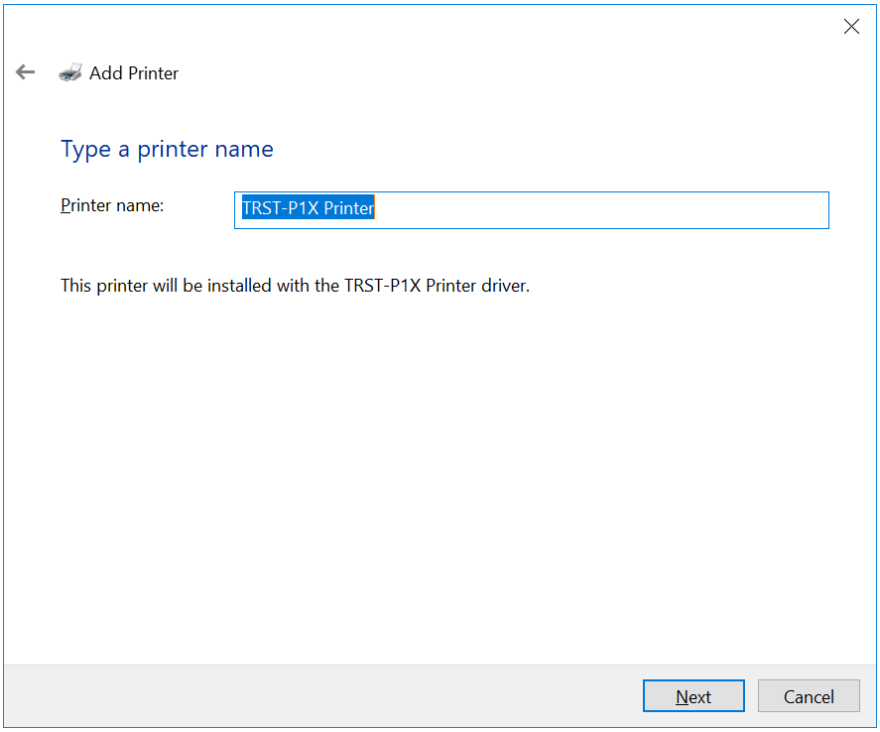


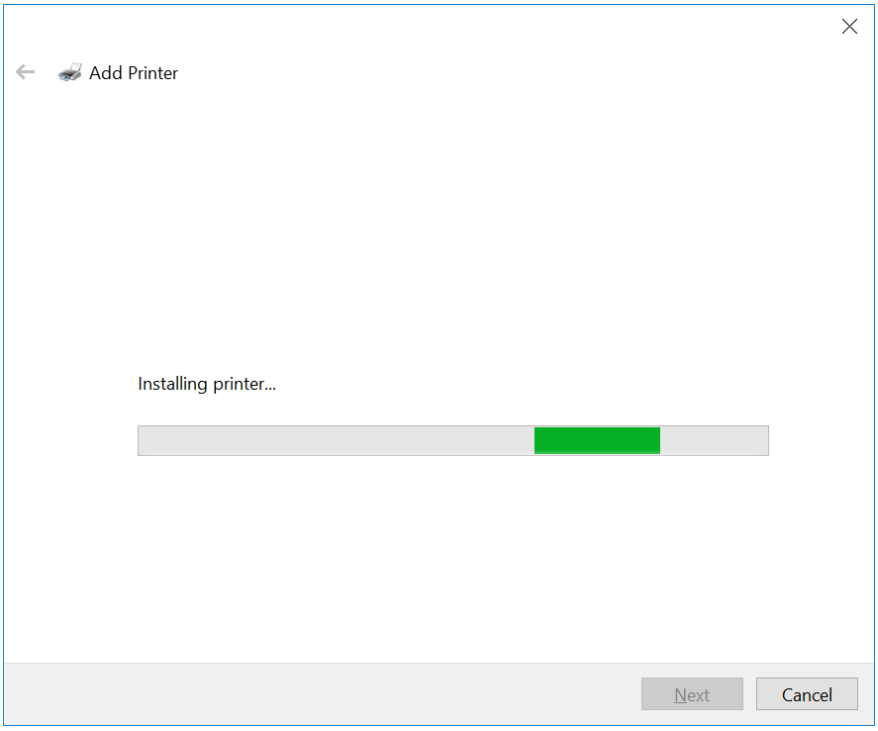
7) Click [Browse…] and find the location of the driver then click [OK].
After install using installer the driver file is located on the following path
C:\Program Files (x86)\TOSHIBA\TECWinDriver
Below is folder location example for TRST-P1X printer



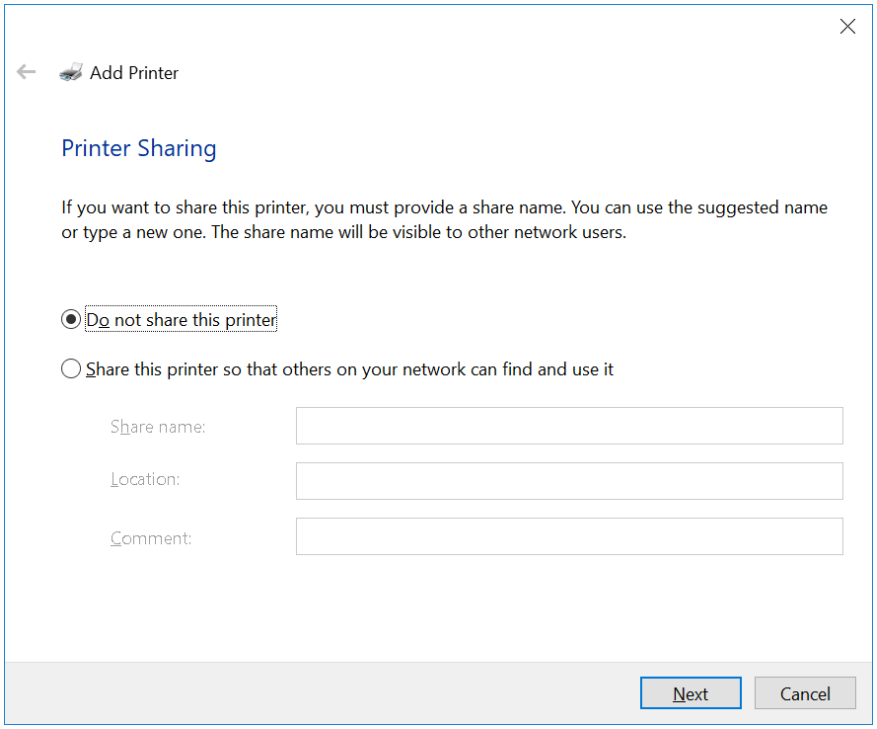
8) The driver name should appear as below. Click [Next] until install.



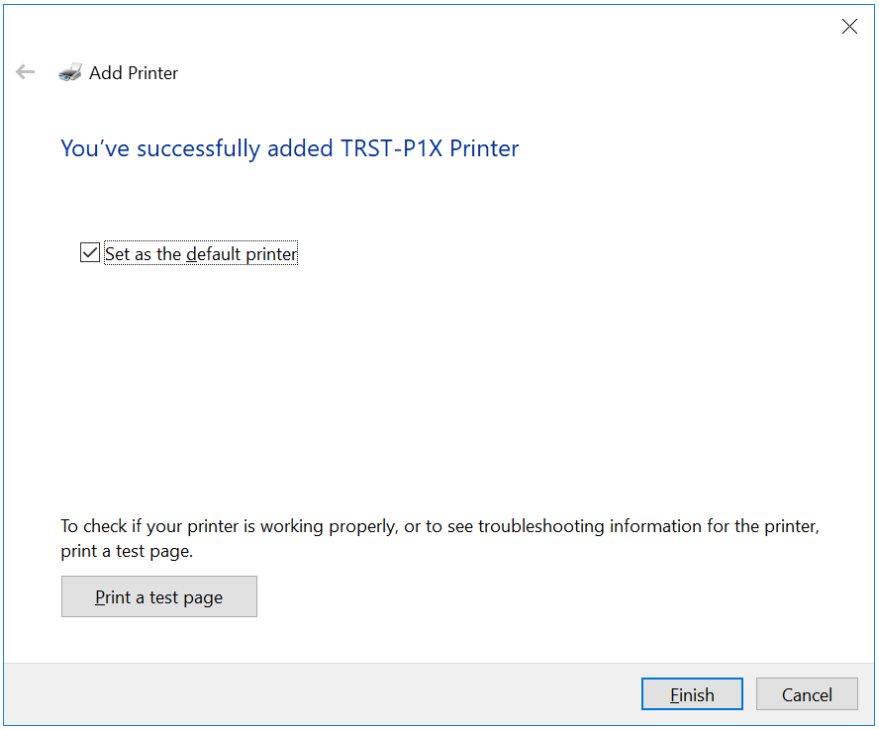

Add Printer
Type a printer name
Printer name:
TRST-P1X Printer
This printer will be installed with the TRST-P1X Printer driver.
Next
Cancel


Add Printer
Installing printer...
Next
Cancel

9) Select desired option then Click [Next]



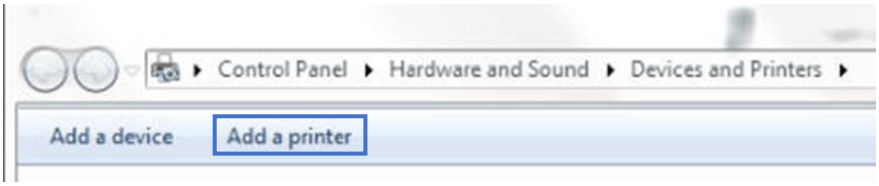
10) Installation complete. Click [Finish]



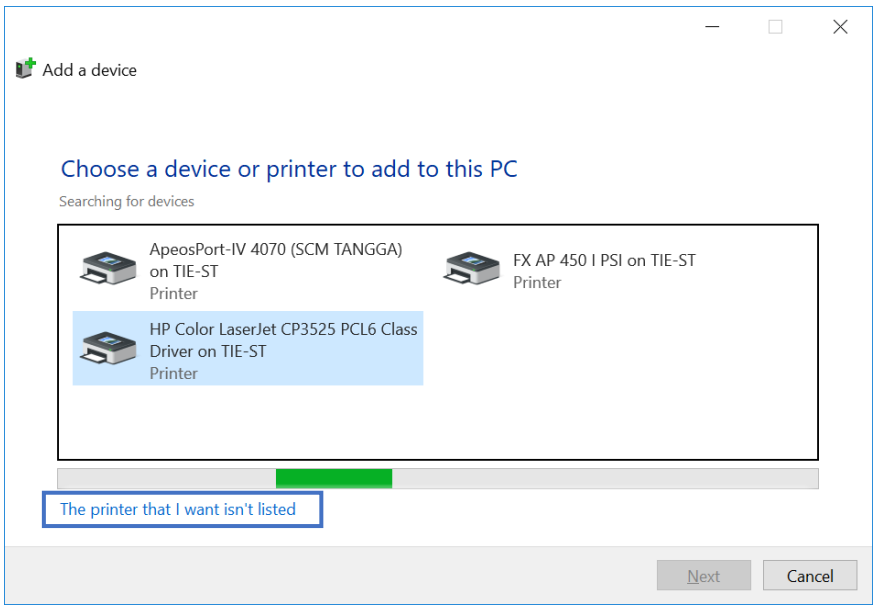
2.2.2 Add Printer with Serial Interface

Below is step to install the driver by using Serial interface

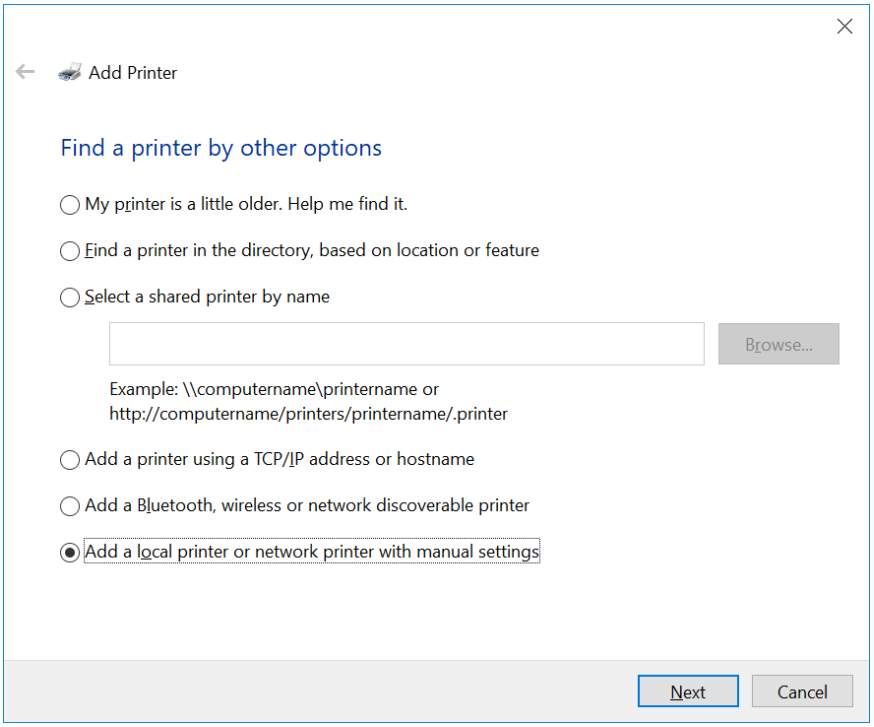
1) Connect printer to PC via RS232 cable
2) Click [Add a printer]



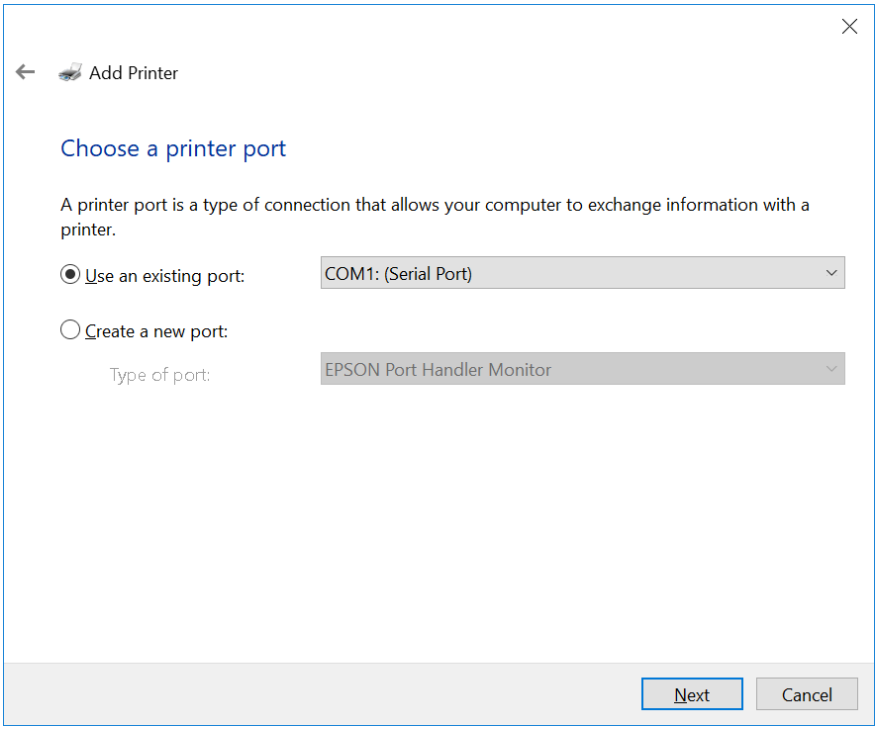
3) Click “The printer that I want isn’t listed”



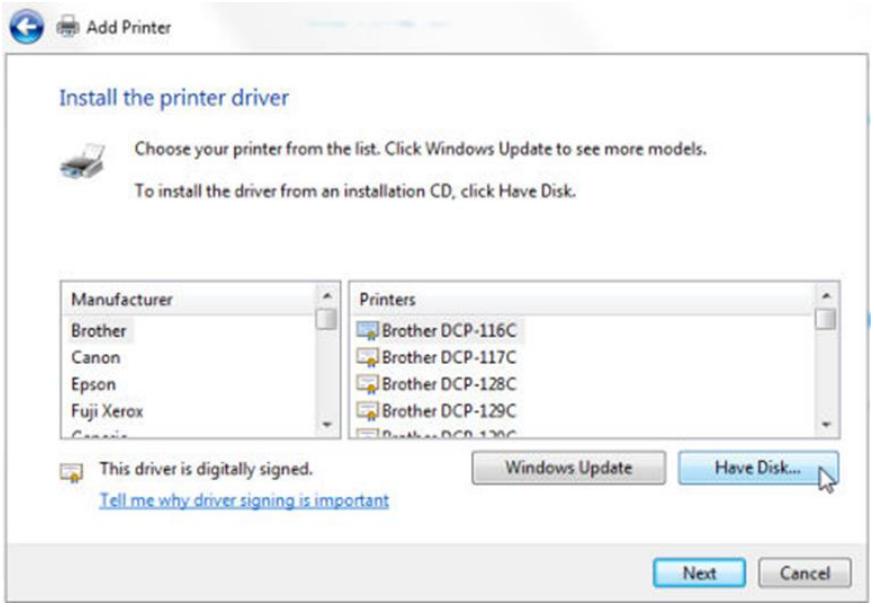
4) Choose “Add a local printer or network printer with manual settings” and click [Next].



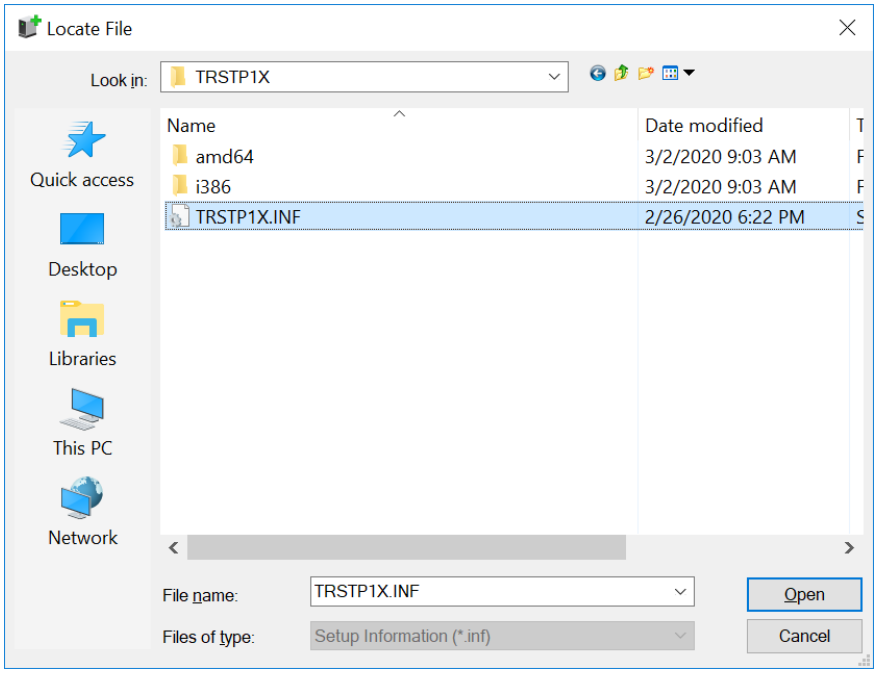
5) Choose a PC COM port that connected to the printer



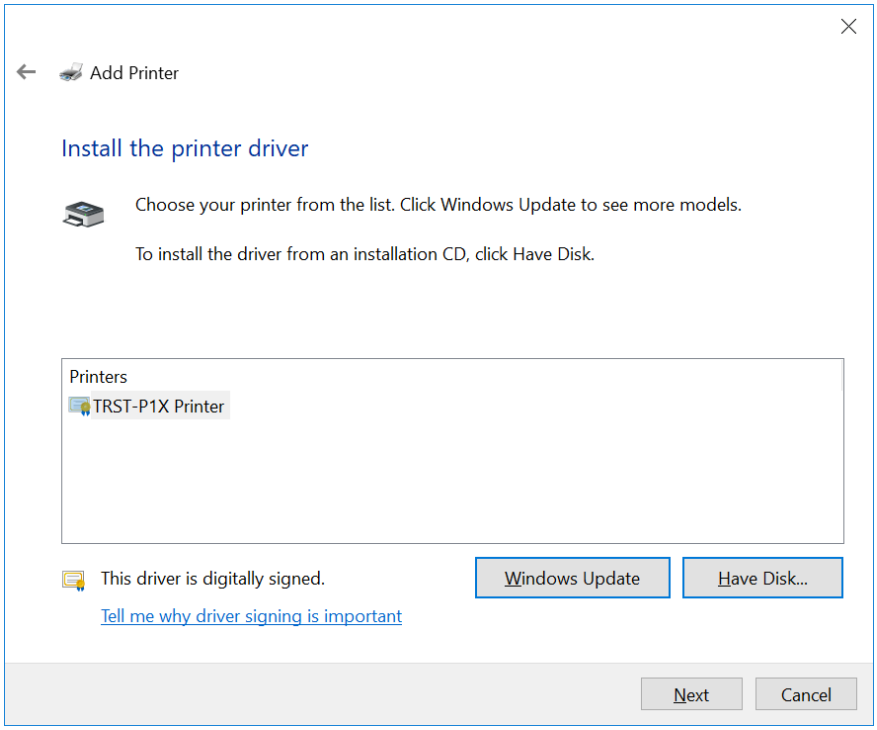
6) Click [Have Disk…]

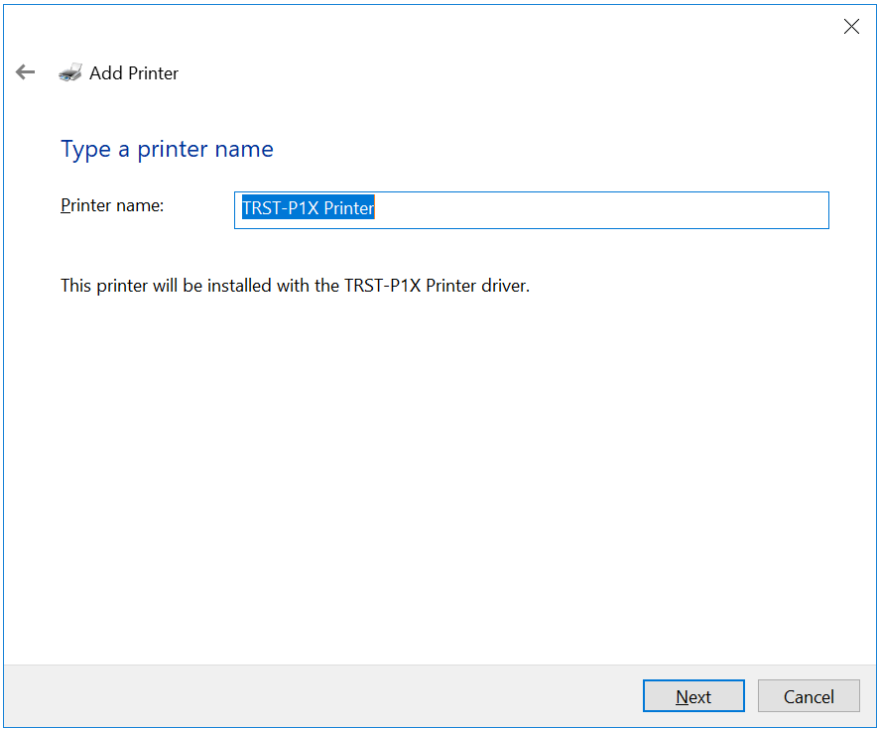


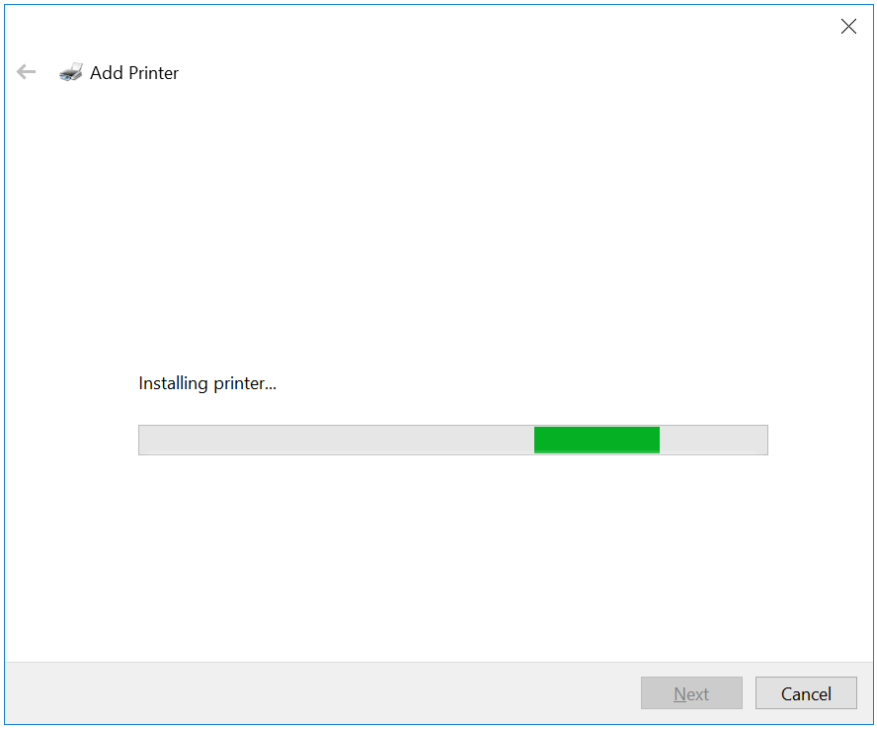
7) Click [Browse…] and find the location of the driver then click [OK]



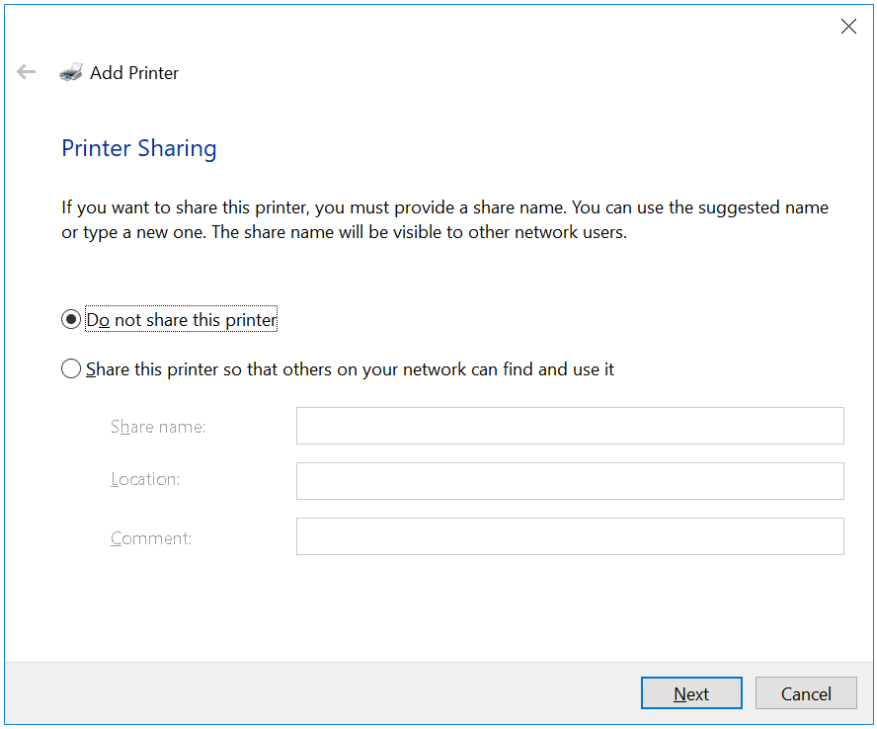
8) The driver name should appear as below. Click [Next] until install.



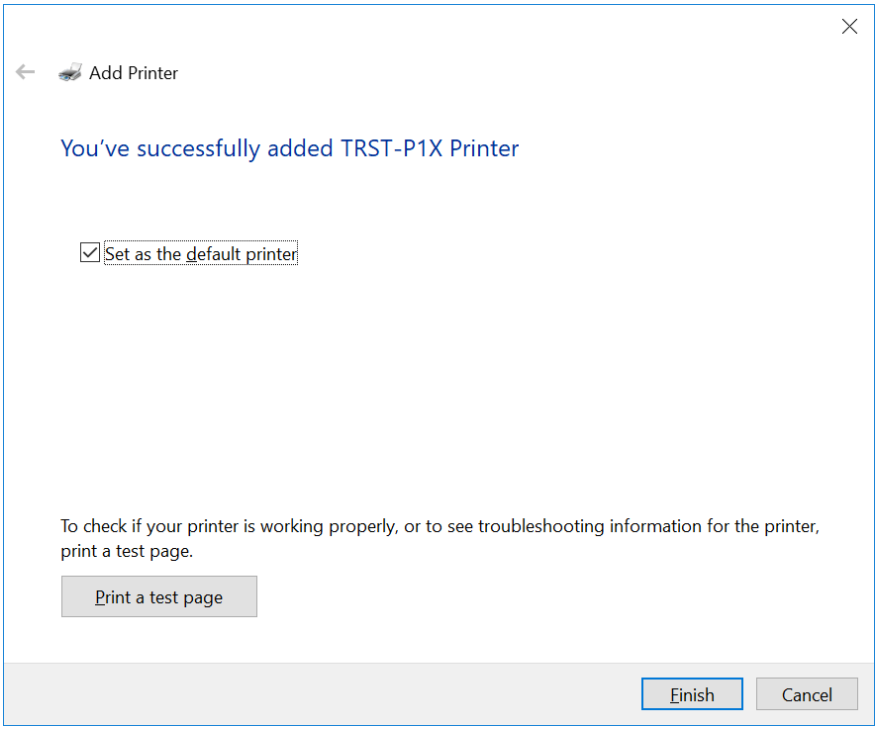

Add Printer
Type a printer name
Printer name:
TRST-P1X Printer
This printer will be installed with the TRST-P1X Printer driver.
Next
Cancel


Add Printer
Installing printer...
Next
Cancel

9) Select desired option then Click [Next]



10) Installation complete. Click [Finish]



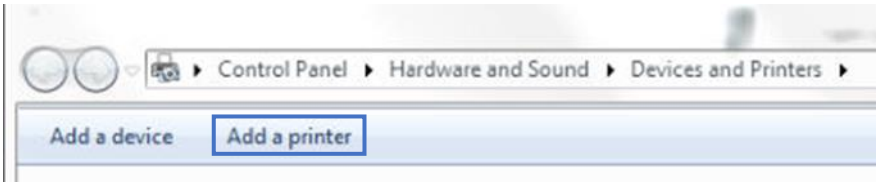
Note:

TOSHIBATEC KOP-3X and TOSHIBATEC KOP-3S06 Printers not support serial interface.

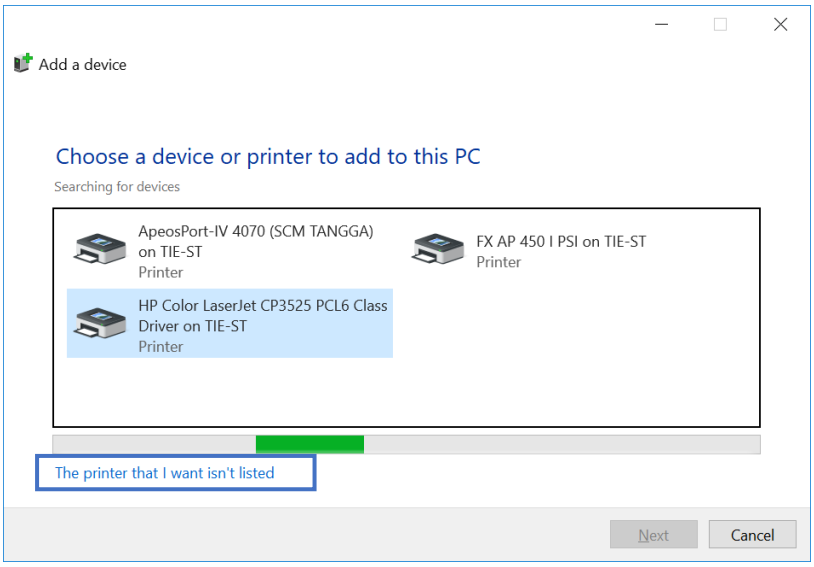
2.2.3 Add Printer with LAN Interface

Below is step to install the driver by using LAN interface

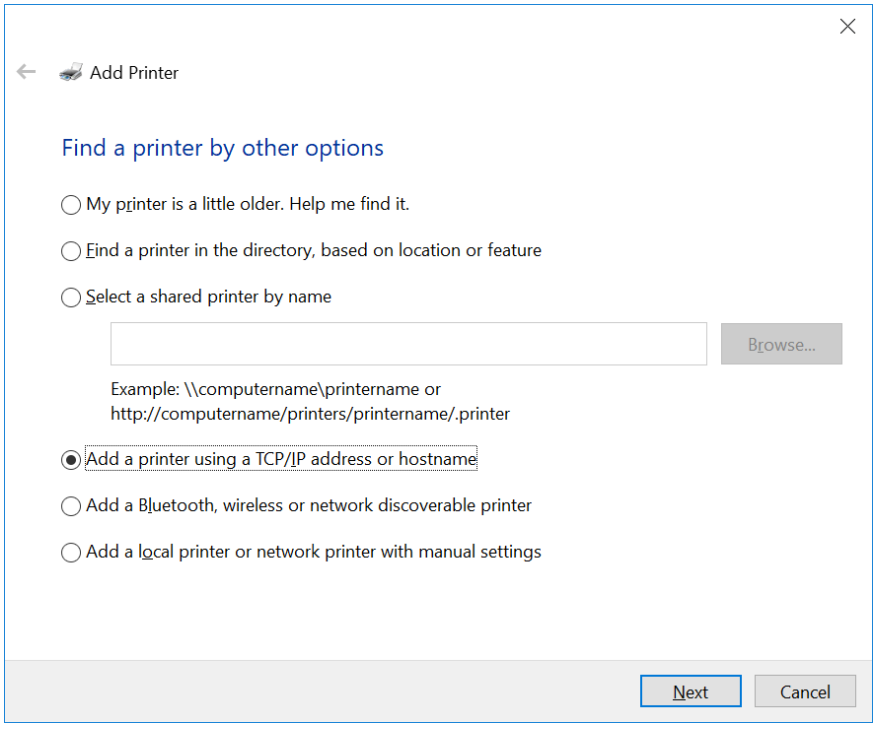
1) Connect printer to PC via LAN cable
2) Click [Add a printer]



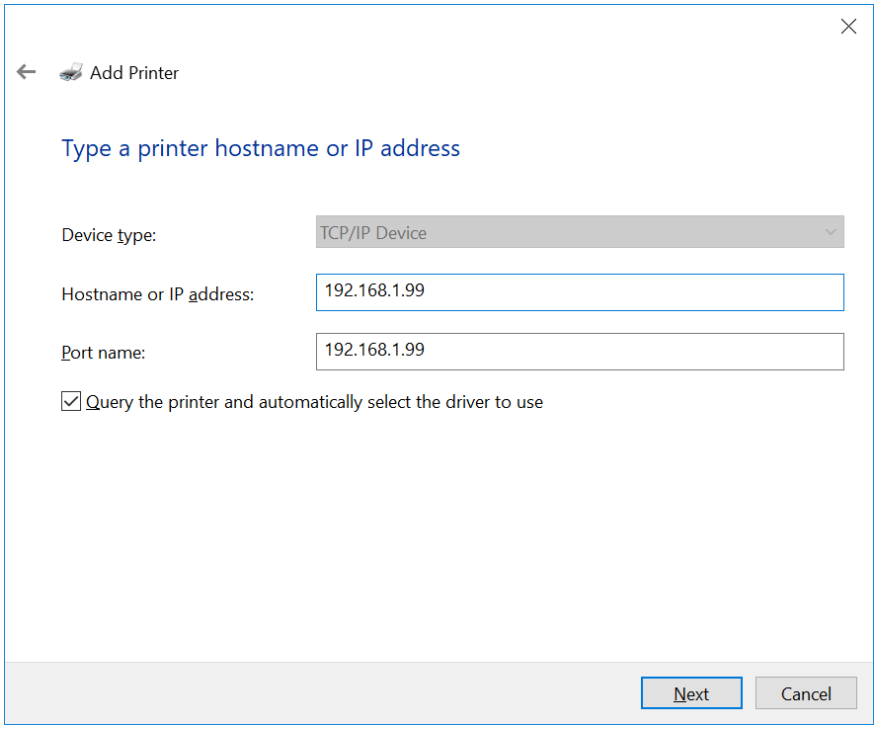
3) Click “The printer that I want isn’t listed”



4) Choose “Add a printer using a TCP/IP address or hostname” and click [Next].

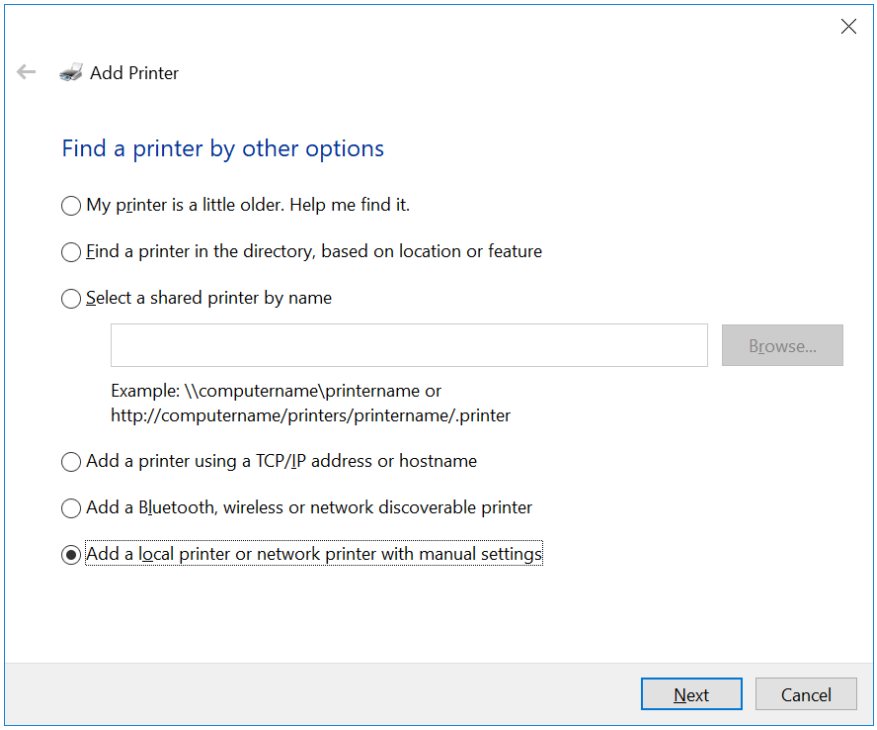


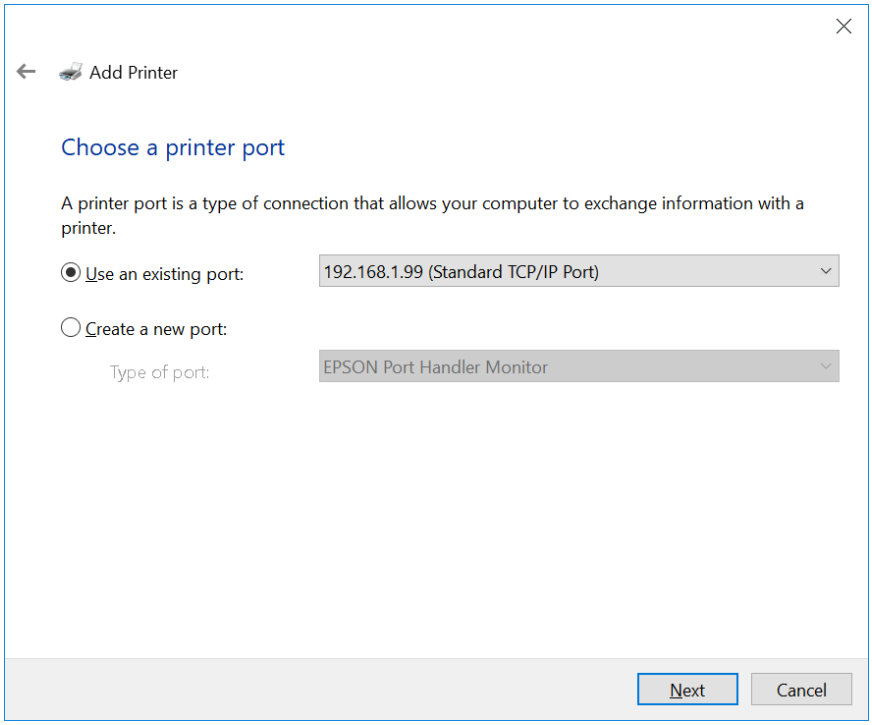
5) Type the Hostname or IP address and the Port name then click [Next]



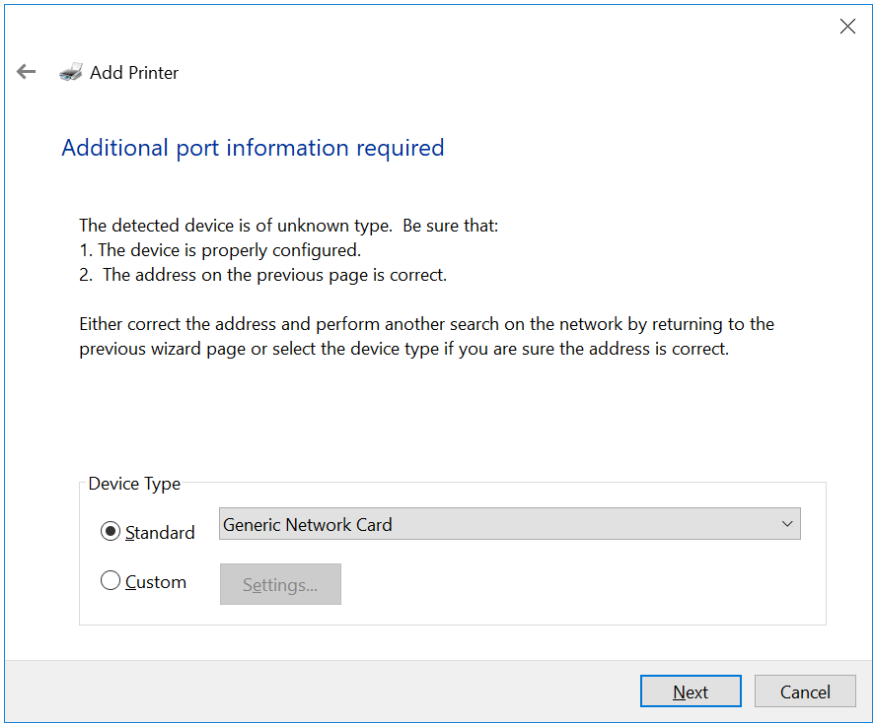
Note:

If TCP/IP Port already added before, select the “Add a local printer or network printer with manual settings” option then select the “Use an existing port” to select the IP address directly

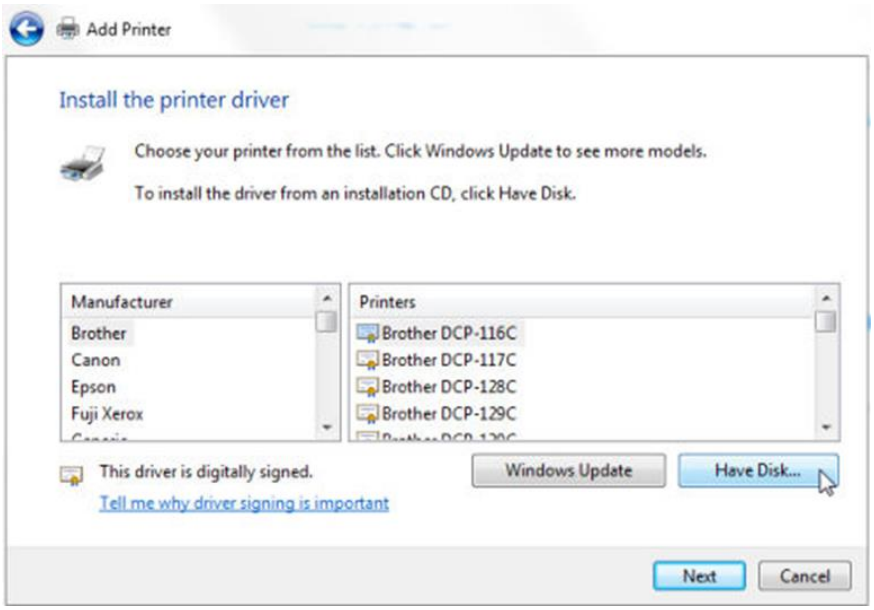




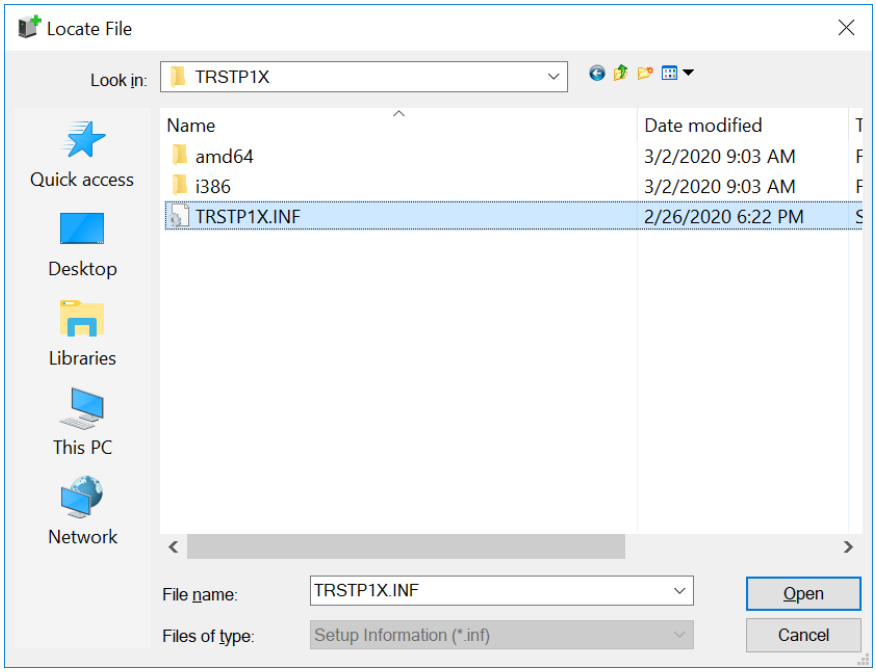
6) Select Device Type then Click [Next]



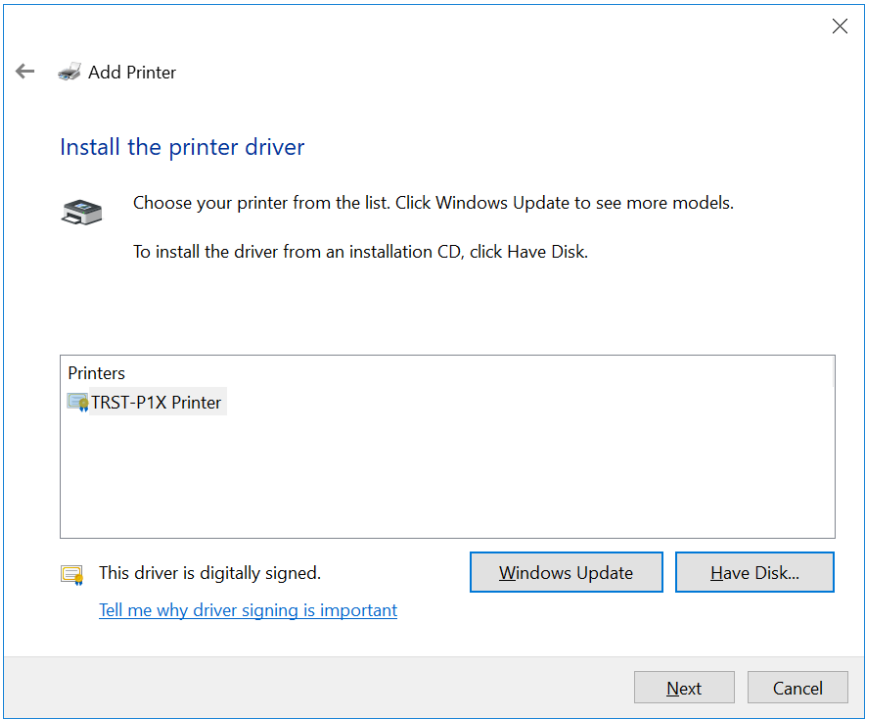
7) Click [Have Disk…]



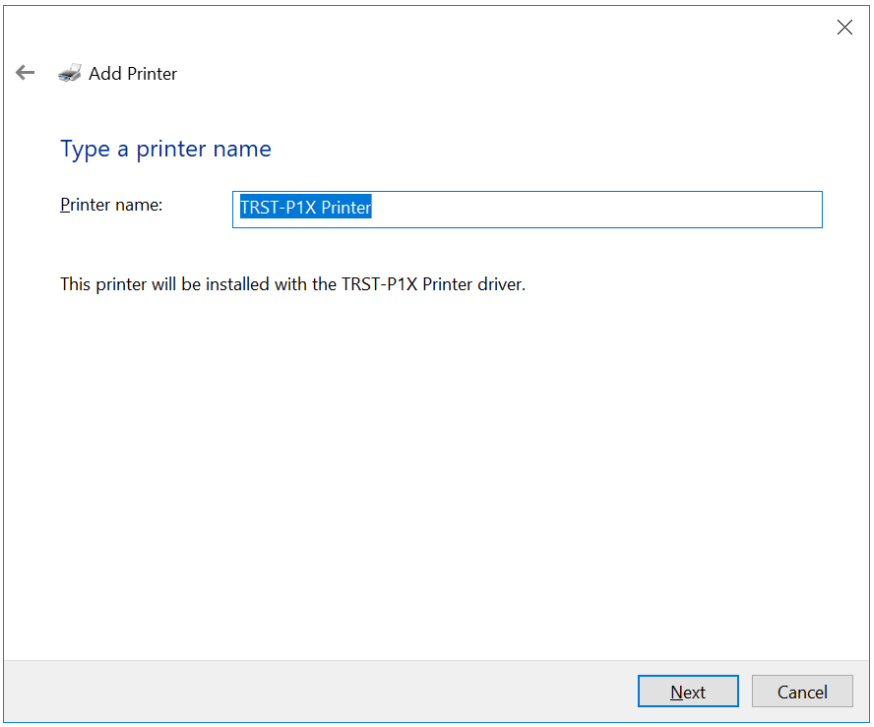
8) Click [Browse…] and find the location of the driver then click [OK]

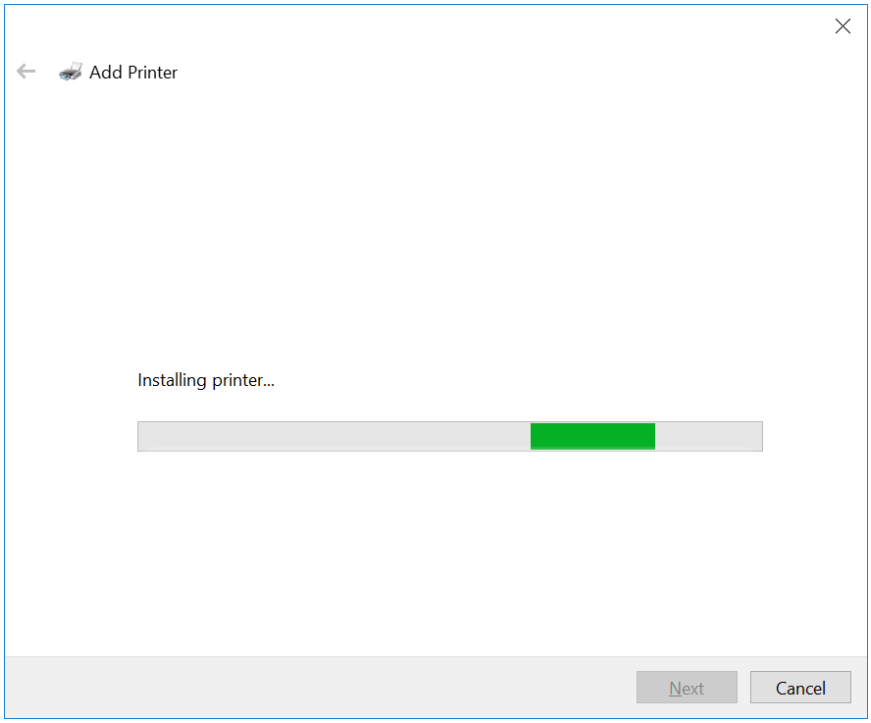


9) The driver name should appear as below. Click [Next] until install.

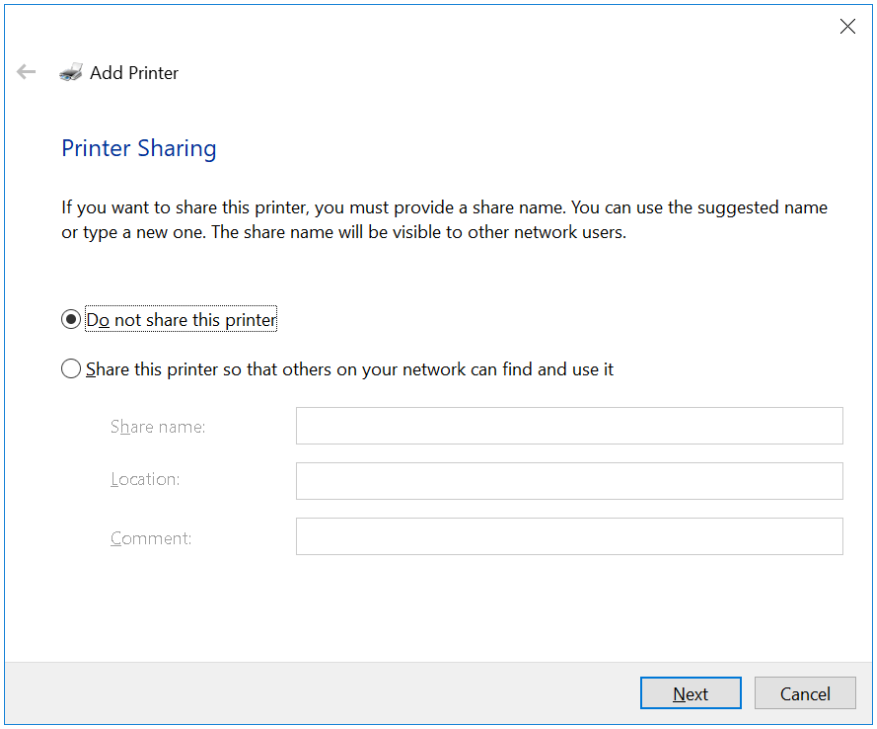


10) Printer name confirmation

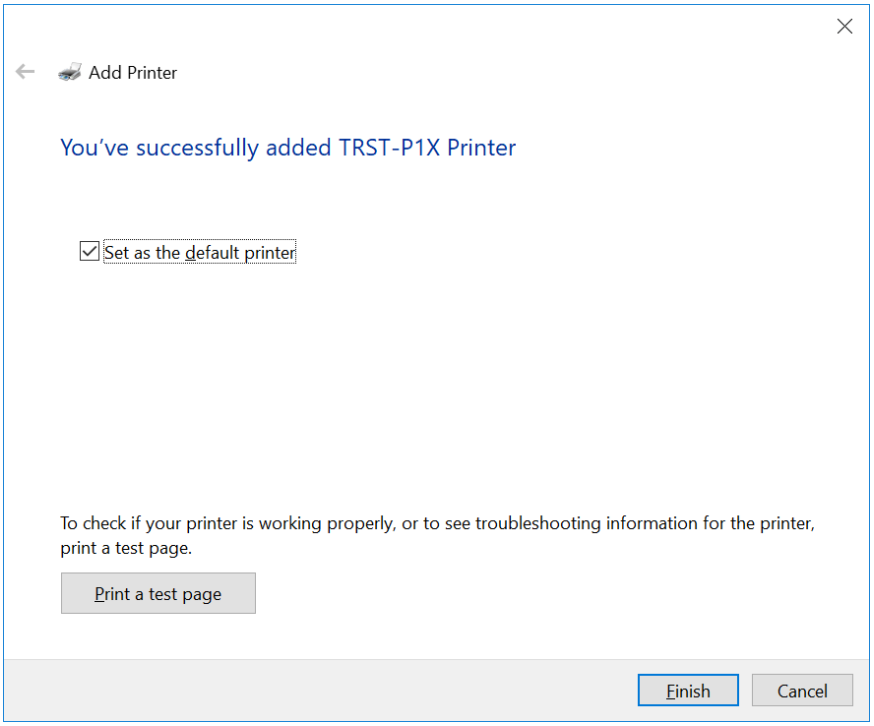




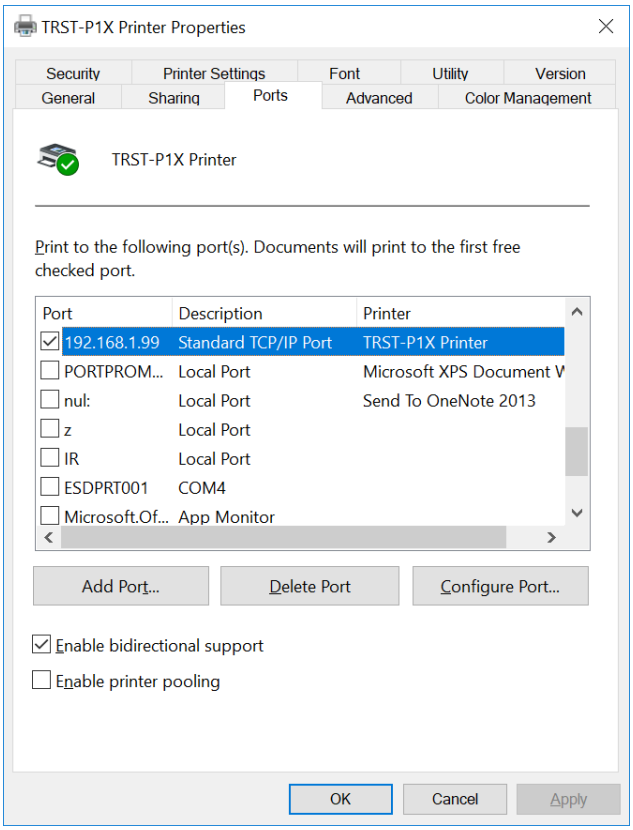
11) Select desired option then Click [Next]



12) Installation complete. Click [Finish]



13) Below is TCP/IP Port creation



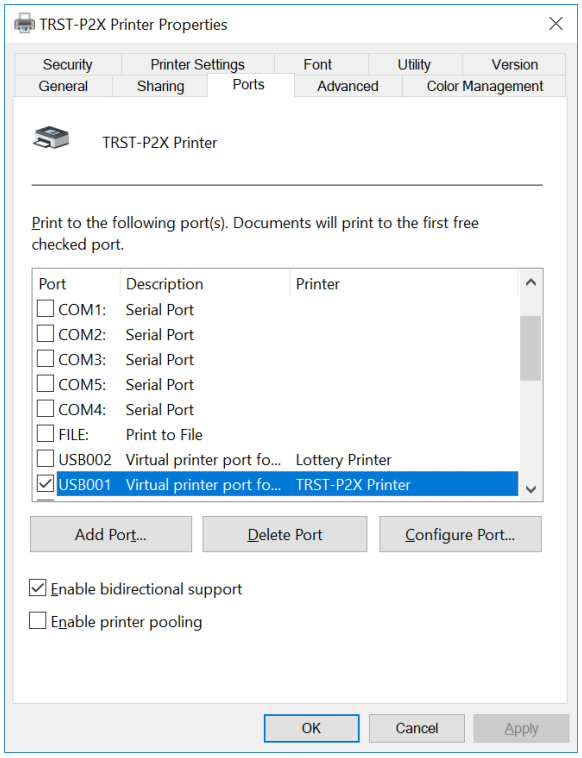
Note:

TOSHIBATEC KOP-3X Printer not support LAN interface.

2.3 Change Printer interface from “Ports” Tab

After installation using installer application, you can change the printer interface from USB interface to RS232 / LAN interface

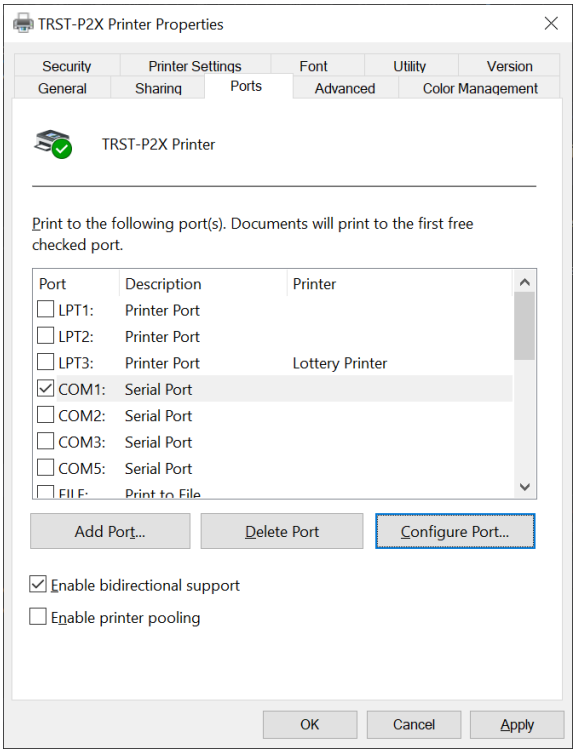
Open Printer Properties then select “Ports” tab



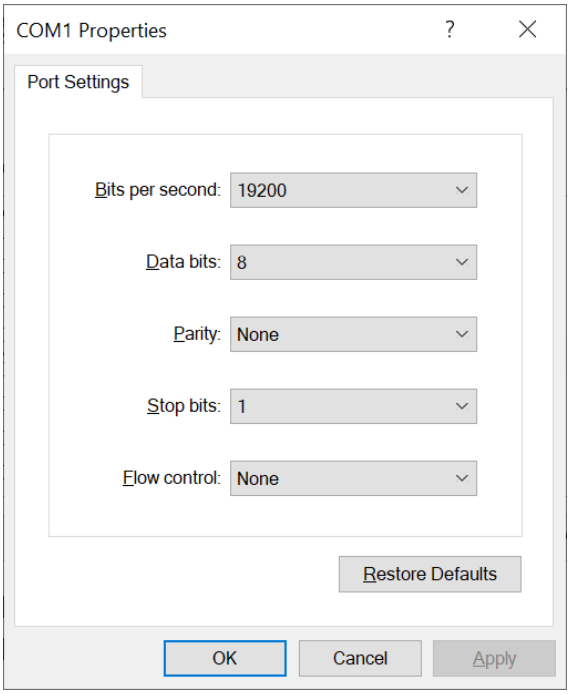
USB to RS232 Interface

Change USB connection to RS232

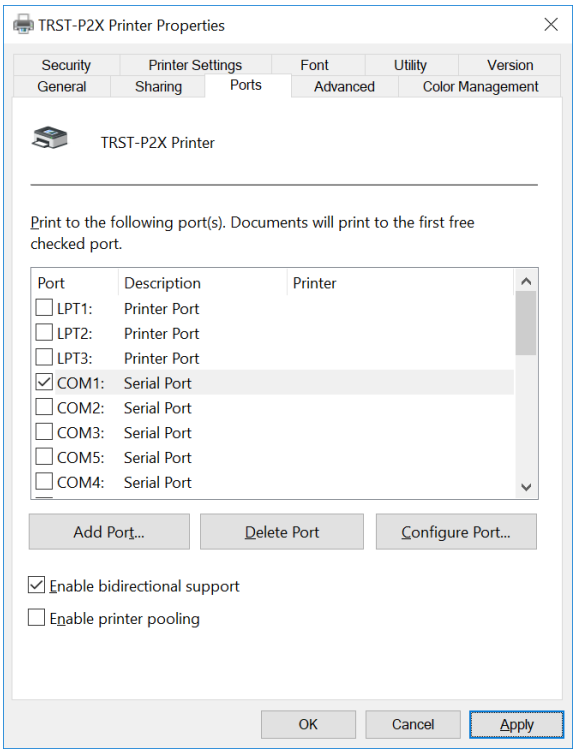
Select COM Port that is connected with the printer. Then Click “Configure Port”



Please select the serial communication parameter based on Printer RS232 Interface Settings on printing configuration. Then click OK Button



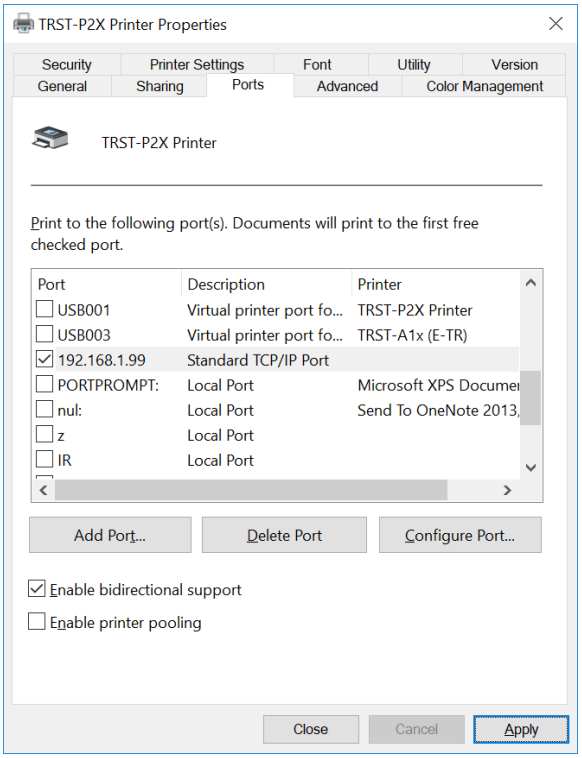
After Port is configured properly then click “Apply” Button



USB to LAN Interface

Change USB connection to LAN

Select LAN Port that connected with the printer. Then Click Apply



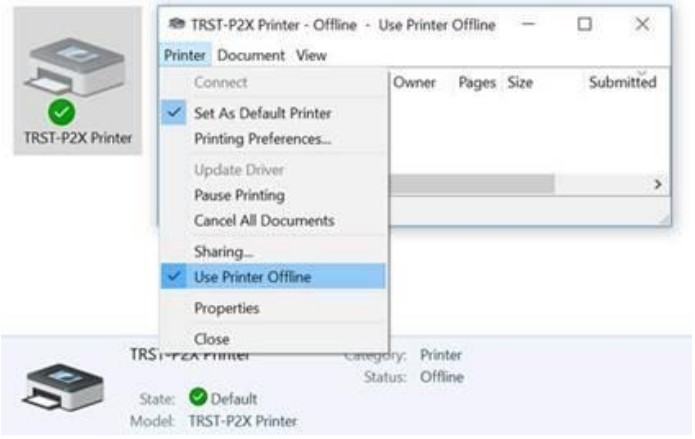
Note:

If printer is in the state "Use Printer Offline", attempting to change the port of the device will successfully change the port, but the printer will continue to be in the offline state.

This case is happen on specific condition as below :

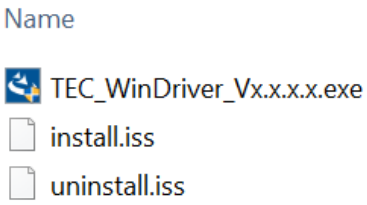
Change port from USB to another interface (RS232/LAN) when USB is disconnect (offline) before open printer properties

In this case need to manually disable the “Use Printer Offline” setting before open Printer properties or after change the port



2.4 Silent Installation

Make sure have the setup.exe and *.iss files:

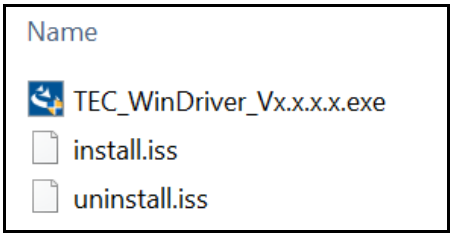


Run silent installation script via command line as the following:

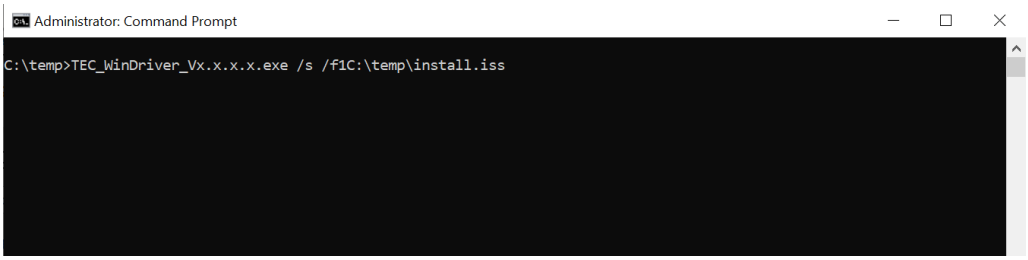
[Setup.exe] /s /f1**[install.iss file path]**

Example:

Installation files location (setup version is Vx.x.x.x) → C:/temp



Silent installation script as the following:

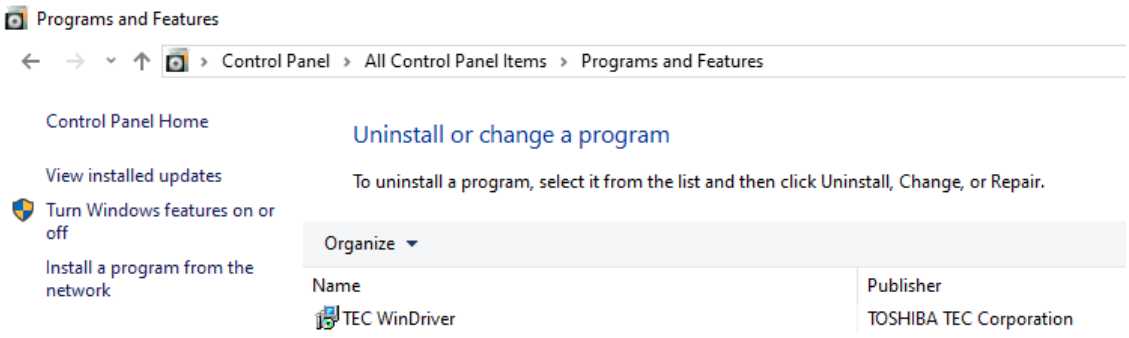


*Note: Please refer **7. Creating *.iss file for driver installation (silent mode)** section to create the **install.iss** file*

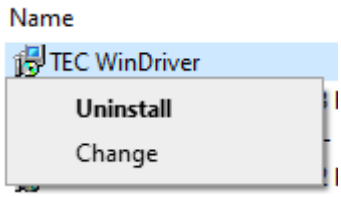
3. How to uninstall the Printer Driver

3.1 Uninstall by using Programs and Features

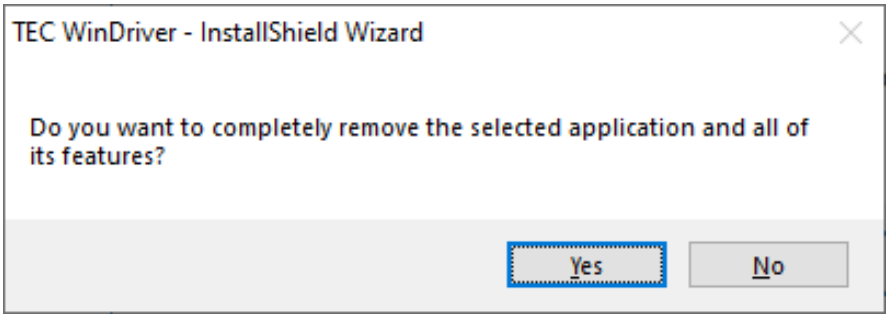
1) Open “Program and Features” window (Control Panel -> Programs -> Programs and Features)



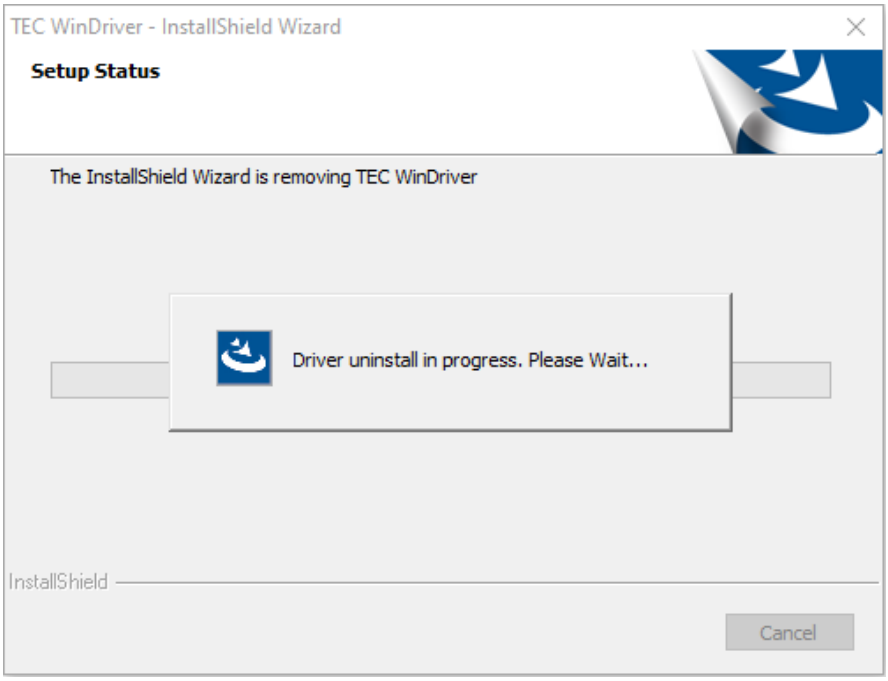
2) Right Click the TEC WinDriver program on “Program and Features” then select “Uninstall”



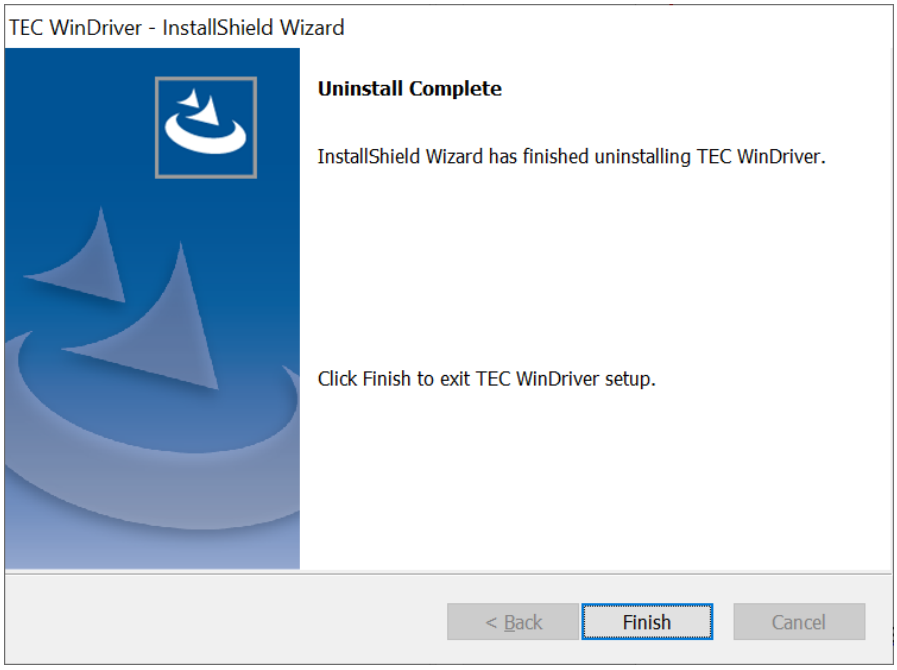
3) Remove application confirmation will appear then click [Yes]



4) Driver Uninstallation Process. Please wait until uninstall finish



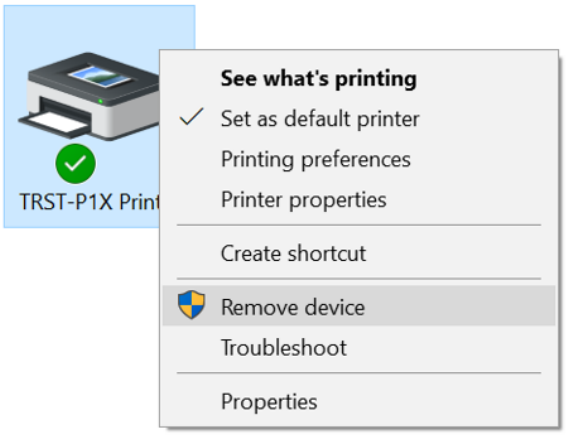
5) Uninstall complete. Click "Finish"



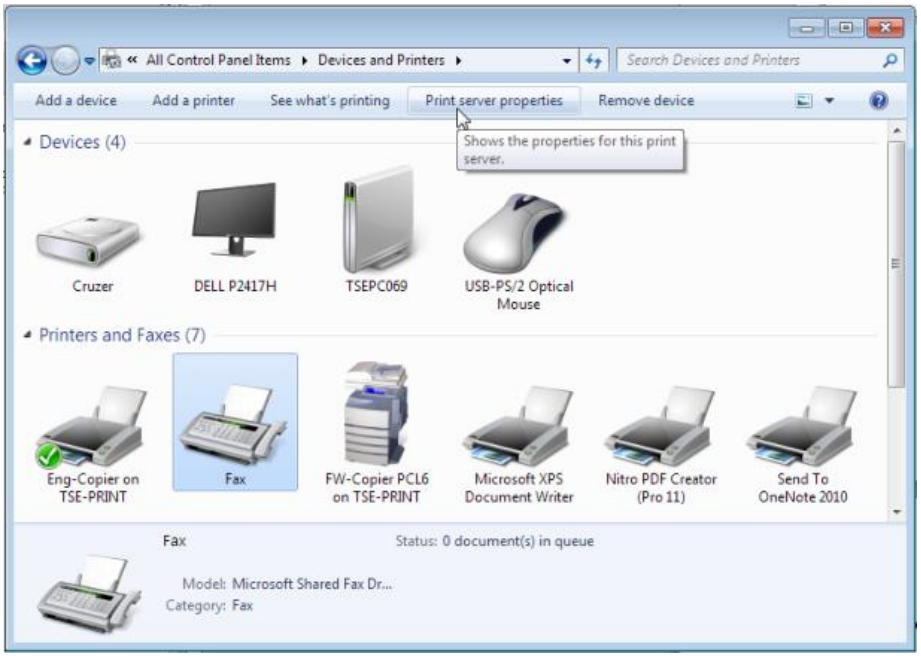
3.2 Manual uninstall by remove device and driver on Print Server Properties

Note: *Ensure Administrator rights to properly uninstall printer driver.*

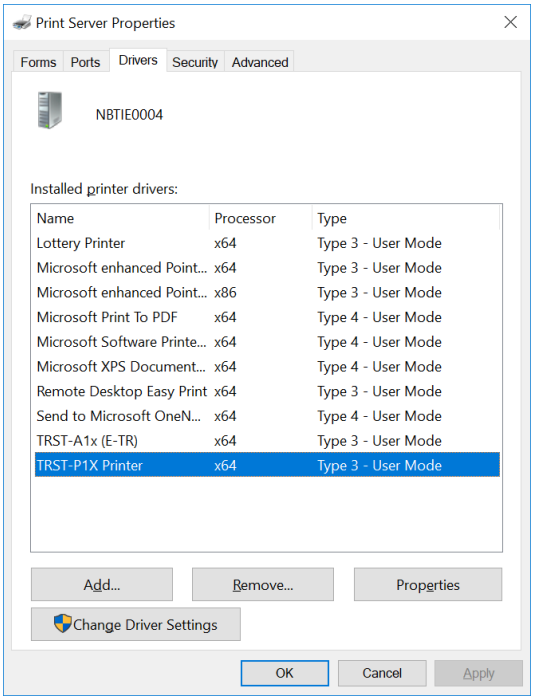
1) Remove device (make sure there are no ongoing print jobs).



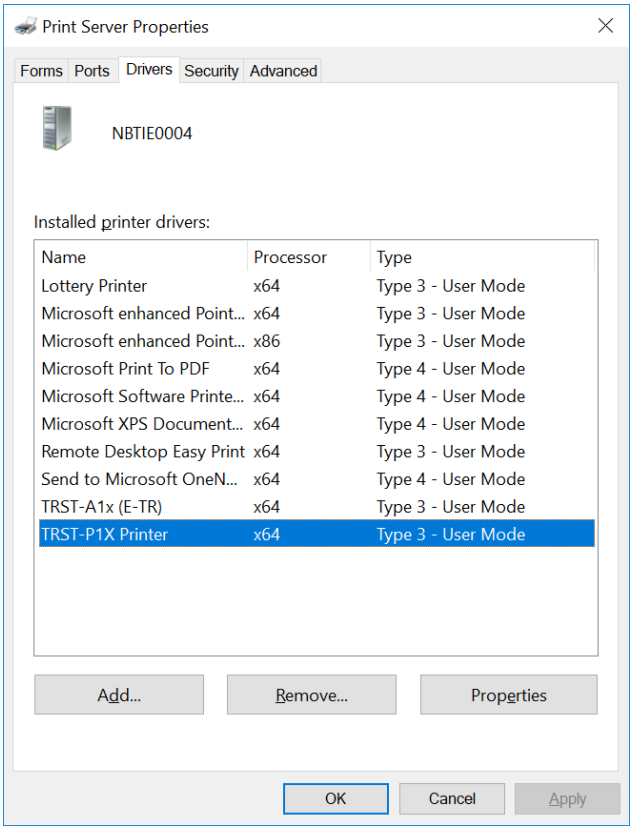
2) Click on any other printer, to see the “Printer server properties”.



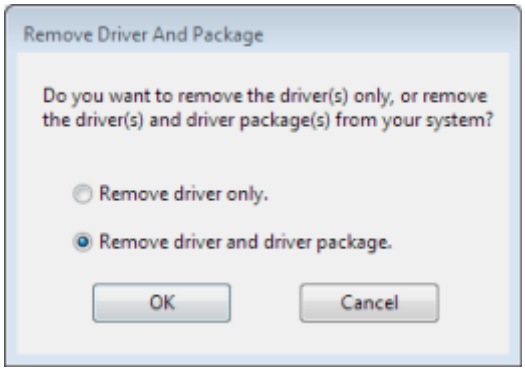
3) In Printer Server Properties, select “change driver settings”



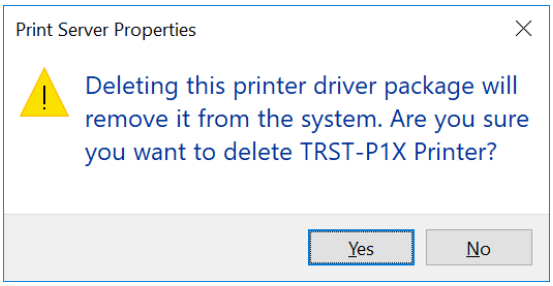
4) Select the driver to be uninstalled and click Remove



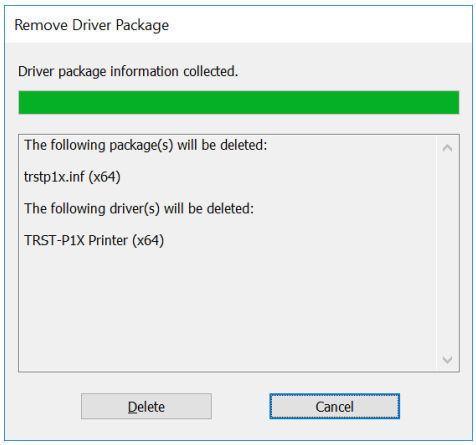
5) Select Remove driver and driver package. Click [OK]



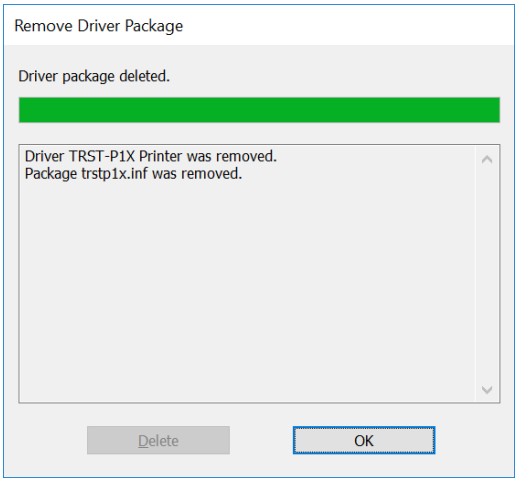
6) Click [Yes]



7) Click [Delete]

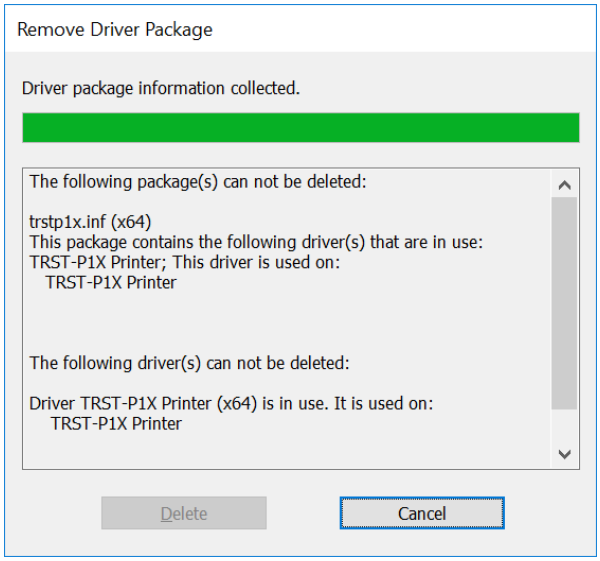


8) Click [Ok]



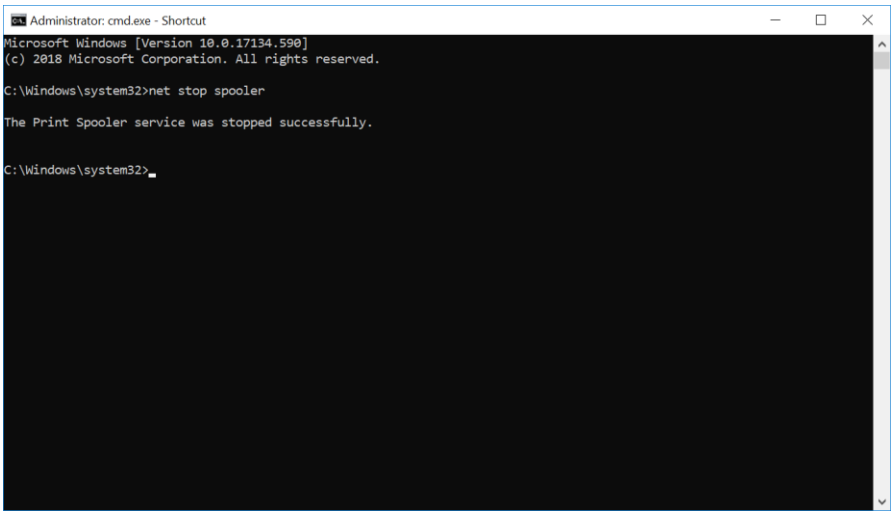
Note:

*If in case the driver cannot be removed, you will see this message:



In this case you need to take the following additional steps:

a. Open the Command Prompt and enter “net stop spooler”.

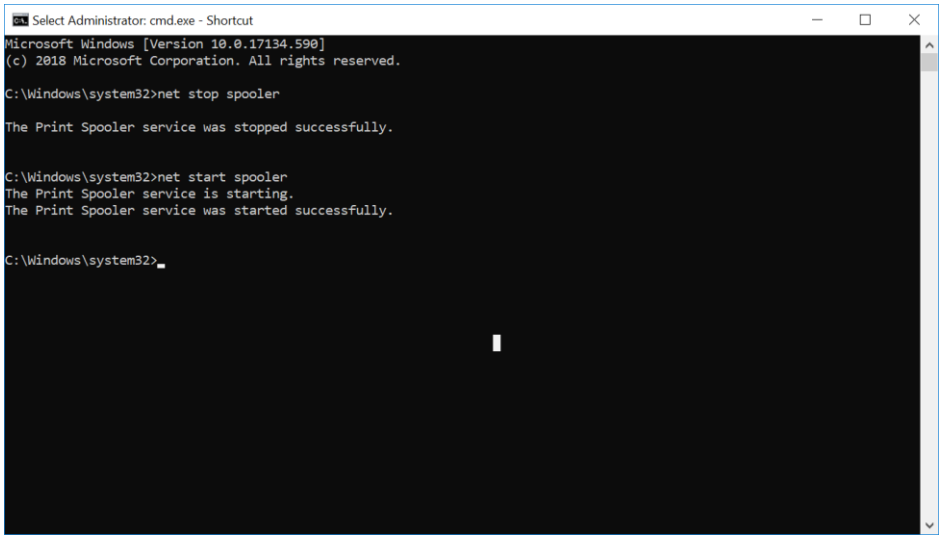


b. Go to C:\Windows\System32 and you will need to delete below file based on installed printer:

Printer	File
TRST-P1X Printer (HSP-150)	TRSTP1XLMN.dll
TRST-P2X Printer (HSP-100)	TRSTP2XLMN.dll
TOSHIBATEC KOP-3X	KOP3LMN.dll
TOSHIBATEC KOP-3S06	KOP3SLMN.dll
TRST-L1X Printer	TRSTL1XLMN.dll

Proceed to delete above file.

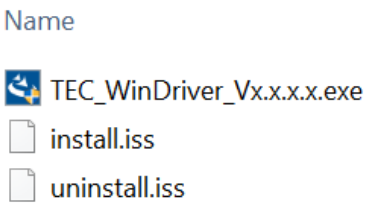
c. Then next enter “net start spooler”.



d. Then go back to Printer Server Properties (step 3) and try to remove again.

3.3 Silent Uninstallation

Make sure have the setup.exe and *.iss files:

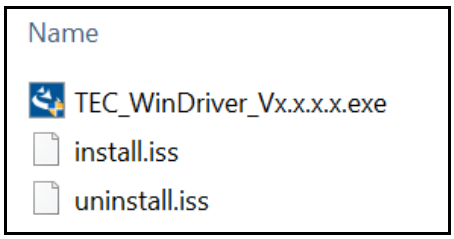


Run silent installation script via command line as the following:

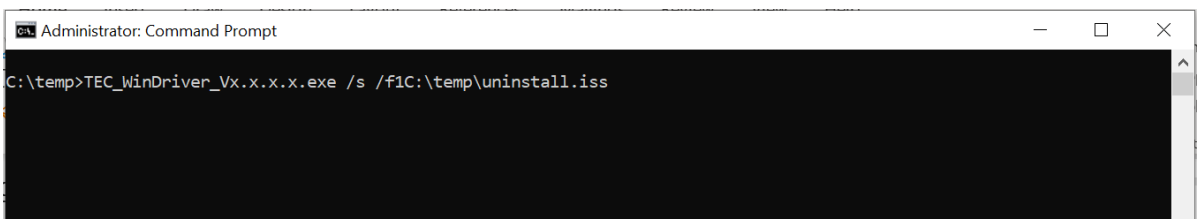
[Setup.exe] /s /f1**[uninstall.iss file path]**

Example:

Installation files location (setup version is Vxx.xx.xx.xx) → C:/temp



Silent uninstallation script as the following:

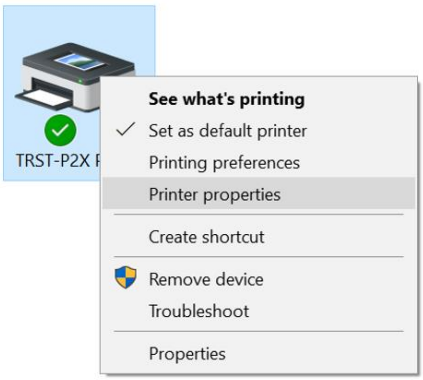


*Note: Please refer **7. Creating *.iss file for driver installation (silent mode)** section to create the **uninstall.iss** file*

4. Printer Properties

Printer Properties could be accessed from the "Devices and Printers".

Right-Click the printer driver name and select Printer properties button

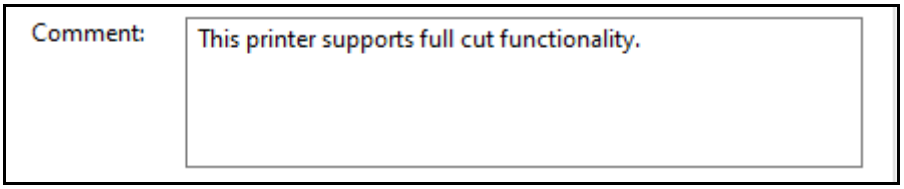


Below table show the property sheets.

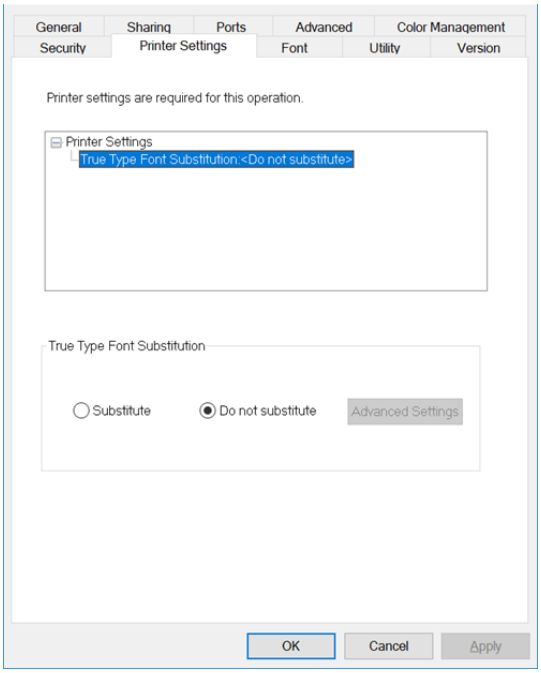
Sheet	Provider	Note
General Sharing Ports Advanced Color Management Security	Windows	
Printer Settings Font Utility Version	Vendor	Created by Printer Driver's Vendor

Note:

For the TRST-P2N2-XX printer model, the "General" tab will display the comment field as: "This printer supports full cut functionality."

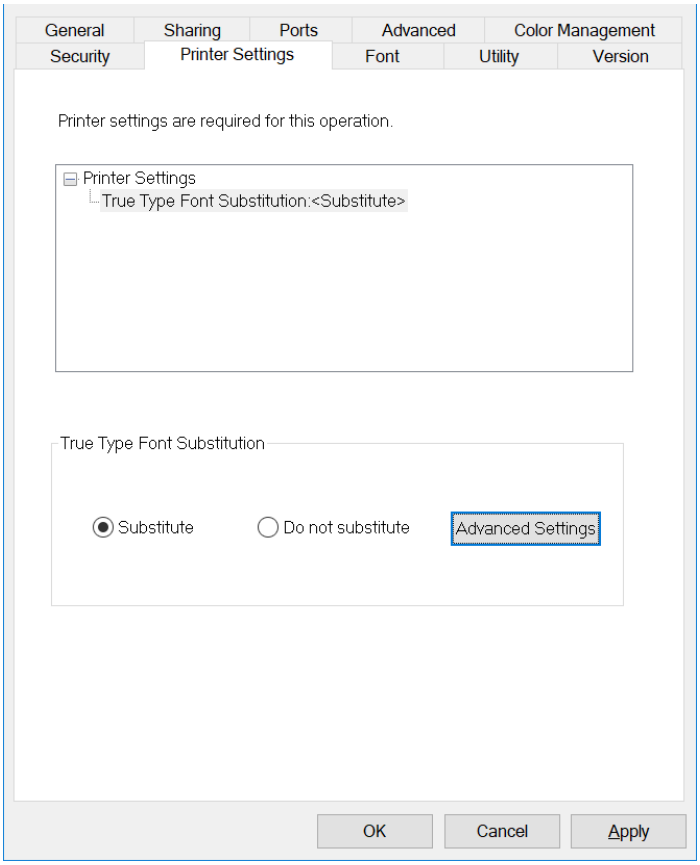


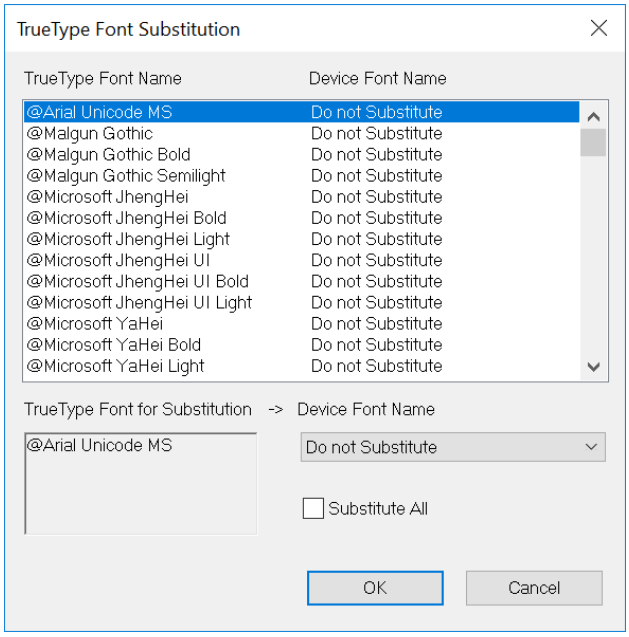
4.1 Printer Settings



True Type Font Substitution

Select **Substitute** option to enable the font substitution feature. Click Advanced Settings to select font type to substitute.





Below is device Font Name list for substitution

- FontA
- FontB
- Control
- controlA
- Barcode(1-8)
- 2D-Code(1-8)

Click OK and the true type font name will substituted with selected device font name

Note:

In case “Substitute All” is checked, the selected "Device Font Name" should be applied to all the True Type Font listed on this dialog (=True Type Font Substitution) and then, all the True Type Font listed should be disabled(=cannot be modified)

FontA & FontB

In case FontA & FontB is selected

There are 2 command sent to printer:

1. **Character font**

   FontA & FontB is character font. Below command will sent when select FontA or FontB

1.7.9 Select character font

ASCII:	ESC M n
Hexadecimal:	1B 4D n
Decimal:	27 77 n
Value of *n*:	0, 48: Font A (Standard pitch font) 1, 49: Font B (Compress pitch font)

2. Character Size

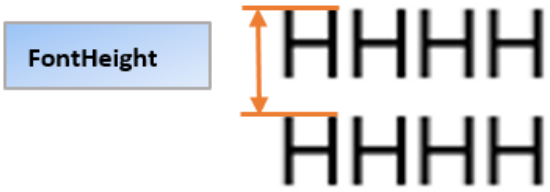
1.7.14	**Select character size**
ASCII:	GS ! n
Hexadecimal:	1D 21 n
Decimal:	29 33 n
Value of *n*:	Bit 0 to 3 = Height magnification (0 – 7) Bit 4 to 7 = Width magnification (0 – 7)
Range of *n*:	0 ≤ n ≤ 7, 16 ≤ n ≤ 23, 32 ≤ n ≤ 39, 48 ≤ n ≤ 55, 64 ≤ n ≤ 71, 80 ≤ n ≤ 87, 96 ≤ n ≤ 103, 112 ≤ n ≤ 119
Default:	0

The character size that sent to printer is automatically assign based on below criteria:

1. FontBaseHeight <= FontHeight < (FontBaseHeight*2) → Font substitute is FontA11 / FontB11
2. (FontBaseHeight*2) <= (FontHeight < FontBaseHeight*4) → Font substitute is FontA22 / FontB22
3. (FontBaseHeight*4) <= (FontHeight < FontBaseHeight*8) → Font substitute is FontA44 / FontB44
4. (FontBaseHeight*8) <= FontHeight → Font substitute is FontA88 / FontB88

Note:

1. **FontBaseHeight : The value is based on Font Size selection on "Font Tab" section (in dots)**
   Font Size1 → FontBaseHeight = 27
   Font Size2 → FontBaseHeight = 34
2. **FontHeight : True Type Font Height (in dots)**



Resolution	180dpi model: 180.0 dpi x 180.0 dpi (7 dots/mm) 203dpi model: 203.2 dpi x 203.2 dpi (8 dots/mm)

3. Font substitute is depend on character font selection (FontA / FontB)

Example:

1. Using AMY Printer, sent document with Arial Font & Size is 11 Points
   Font Size is Font Size1 and Font Substitute setting is FontA
   FontBaseHeight = 27 (Font Size1 Parameter)
   FontHeight = ±4.4mm = 4.4 * 8 = 35 dots (**Note** : 203dpi model -> 8dots/mm)
   Check Criteria :

1. FontBaseHeight <= FontHeight < (FontBaseHeight*2) → Font substitute is FontA11 / FontB11
2. (FontBaseHeight*2) <= (FontHeight < FontBaseHeight*4) → Font substitute is FontA22 / FontB22
3. (FontBaseHeight*4) <= (FontHeight < FontBaseHeight*8) → Font substitute is FontA44 / FontB44
4. (FontBaseHeight*8) <= FontHeight → Font substitute is FontA88 / FontB88

Criteria to use is **No 1**

FontBaseHeight <= FontHeight < (FontBaseHeight*2)

27 <= 35 < 54

Font Substitute setting is FontA so the **Character size** is use **FontA11**

2. Using CLARA Printer, sent document with Calibri Font & Size is 20 Points

Font Size is Font Size1 and Font Substitute setting is FontB

FontBaseHeight = 27 (Font Size1 Parameter)

FontHeight = ±8.7mm = 8.7 * 7 = 61 dots (**Note** : 180dpi model -> 7dots/mm)

Check Criteria :

1. FontBaseHeight <= FontHeight < (FontBaseHeight*2) → Font substitute is FontA11 / FontB11
2. (FontBaseHeight*2) <= (FontHeight < FontBaseHeight*4) → Font substitute is FontA22 / FontB22
3. (FontBaseHeight*4) <= (FontHeight < FontBaseHeight*8) → Font substitute is FontA44 / FontB44
4. (FontBaseHeight*8) <= FontHeight → Font substitute is FontA88 / FontB88

Criteria to use is **No 2**

(FontBaseHeight*2) <= FontHeight < (FontBaseHeight*4)

54 <= 61 < 108

Font Substitute setting is FontB so the **Character size** is use **FontB22**

Note:

FontA11 / FontB11 = Character width & height size is 1 Time (normal) -> Command to sent : 1D 21 00

FontA22 / FontB22 = Character width & height size is 2 Times -> Command to sent : 1D 21 11

FontA44 / FontB44 = Character width & height size is 4 Times -> Command to sent : 1D 21 33

FontA88 / FontB88 = Character width & height size is 8 Times -> Command to sent : 1D 21 77

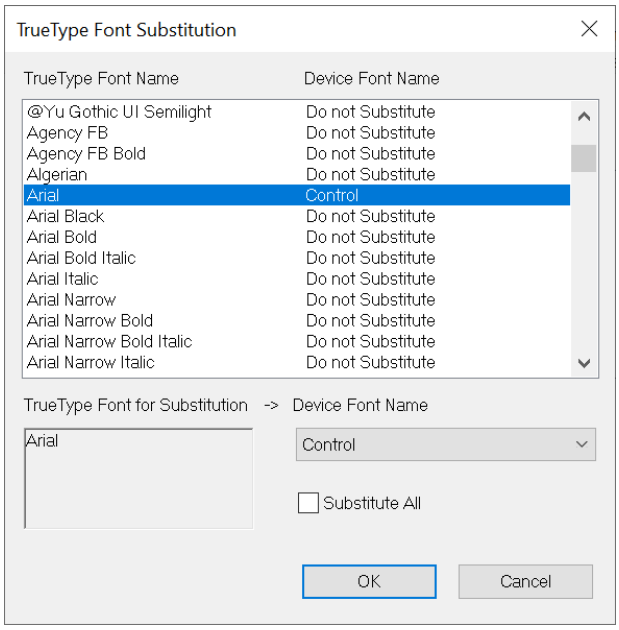
Control Font

Driver convert character data to control code based on table below

Character code	Explanation	Sending command (Hex decimal)
5	Send HT	09
6	Send LF	0A
7	Send CR	0D
a	Drawer open 50ms	1B 70 00 19 FF
b	Drawer open 100ms	1B 70 00 32 FF
c	Drawer open 150ms	1B 70 00 4B FF
d	Drawer open 200ms	1B 70 00 64 FF
e	Drawer open 250ms	1B 70 00 7D FF
g	Non feed paper + Partial cut	1D 56 01
w	Positioning adjustment Left	1B 61 00
x	Positioning adjustment Center	1B 61 01
y	Positioning adjustment Right	1B 61 02
A	Drawer open 50ms	1B 70 00 19 FF
B	Drawer open 100ms	1B 70 00 32 FF
C	Drawer open 150ms	1B 70 00 4B FF
D	Drawer open 200ms	1B 70 00 64 FF
E	Drawer open 250ms	1B 70 00 7D FF
P	Paper feeding + Partial cut	1D 56 42 00

Example:

1. Set Font Substitution setting for Arial to Control font



- Open Microsoft Word document then set the font to Arial
- Connect cash drawer to the printer
- Type “g” then perform printing

Result : Cash Drawer will open with pulse on time 50ms

Note:

TOSHIBATEC KOP-3X and TOSHIBATEC KOP-3S06 Printers not support cash drawer.

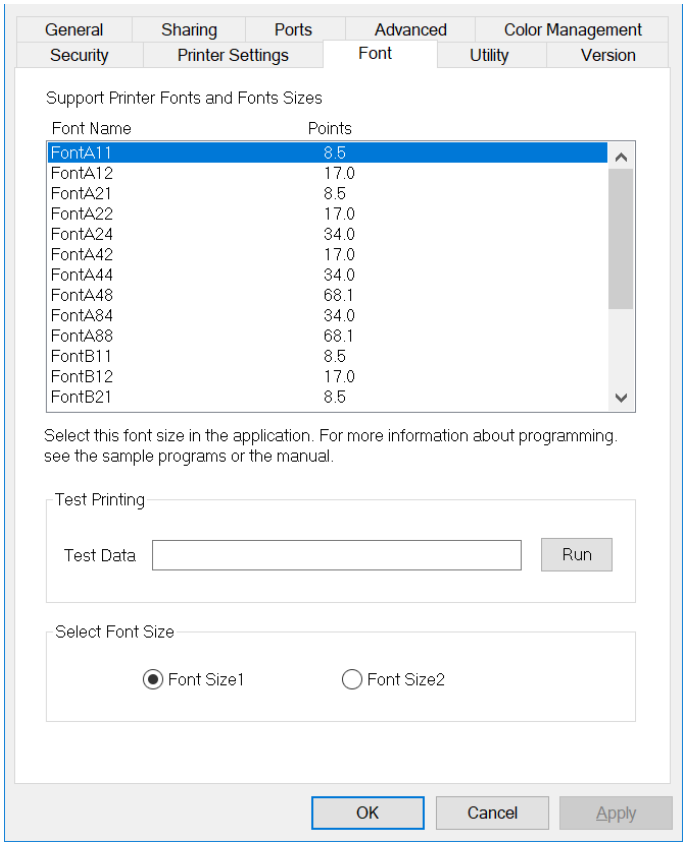
ControlA Font

Driver convert character data to control code based on “Edit ControlA Font” Setting on “Utility” Tab then send command to the printer. (Please see Edit ControlA Section for the example to use ControlA Font)

4.2 Font

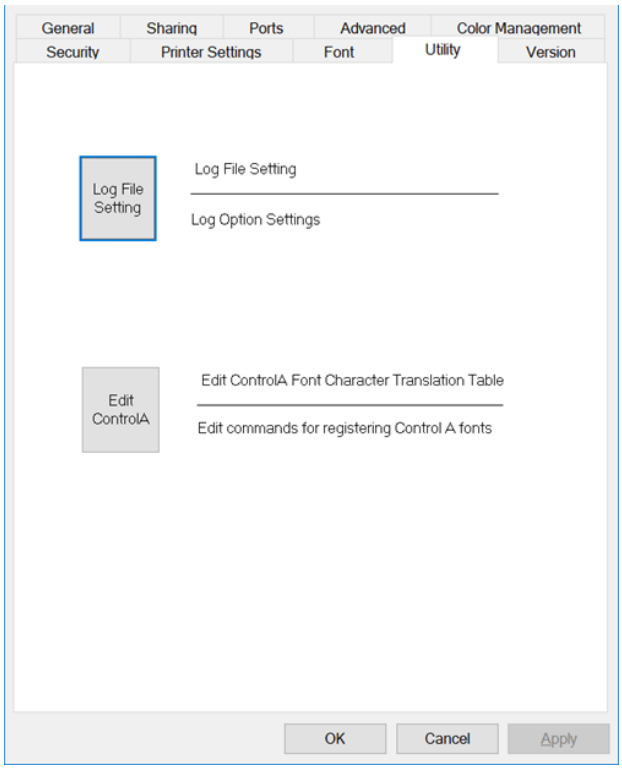
Select the printer font and send the character data to the printer. Below are printer fonts that can be selected by an application. The sizes of those fonts are fixed.

Font1				Font2			
Font Name	Point	Font Name	Point	Font Name	Point	Font Name	Point
FontA11	8.5	FontB11	8.5	FontA11	9.9	FontB11	9.9
FontA12	17.0	FontB12	17.0	FontA12	19.9	FontB12	19.9
FontA21	8.5	FontB21	8.5	FontA21	9.9	FontB21	9.9
FontA22	17.0	FontB22	17.0	FontA22	19.9	FontB22	19.9
FontA24	34.0	FontB24	34.0	FontA24	39.7	FontB24	39.7
FontA42	17.0	FontB42	17.0	FontA42	19.9	FontB42	19.9
FontA44	34.0	FontB44	34.0	FontA44	39.7	FontB44	39.7
FontA48	68.1	FontB48	68.1	FontA48	79.4	FontB48	79.4
FontA84	34.0	FontB84	34.0	FontA84	39.7	FontB84	39.7
FontA88	68.1	FontB88	68.1	FontA88	79.4	FontB88	79.4

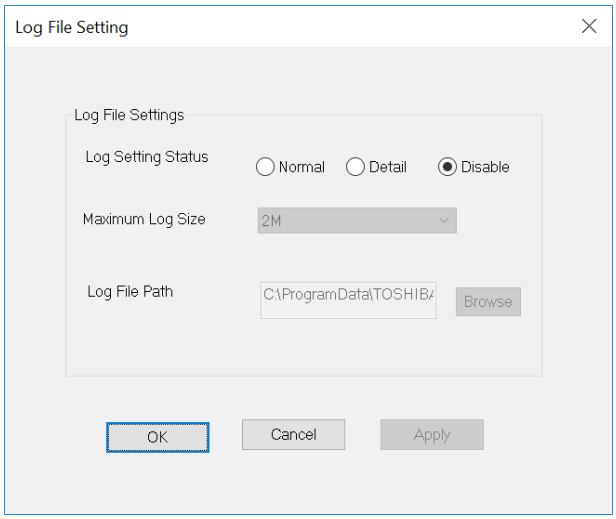


1. **Test Data** : Any string can be input in this EditBox. (Example: ABCD1234)
2. **Run** : On clicking this button, string in "Test Data" is sent to the printer by generating printing job
3. **Select Font Size** : Select Font Size1 / Font Size2 used on printer

4.3 Utility



4.3.1 Log File Setting



This dialog is to configure Log File Setting

1. **Log Setting Status**

   The default value is Disable. To enable the log please select Normal / Detail.

   If selected setting is Normal / Detail printer driver will save log into file.

2. Maximum Log Size

Below are default value

Normal : 2M

Detail : 10M

Each file size of the log is 1M Bytes and the file count is based on Maximum Log Size Setting.

3. Log File Path

Default location for the log file

TRST-P1X / HSP-150 = “C:\ProgramData\TOSHIBA\TRSTP1X\logs”

TRST-P2X / HSP-100 = “C:\ProgramData\TOSHIBA\TRSTP2X\logs”

TOSHIBATEC KOP-3X = “C:\ProgramData\TOSHIBA\KOP-3X\logs”

TOSHIBATEC KOP-3S06 = “C:\ProgramData\TOSHIBA\KOP-3S06\logs”

TRST-L1X = “C:\ProgramData\TOSHIBA\TRSTL1X\logs”

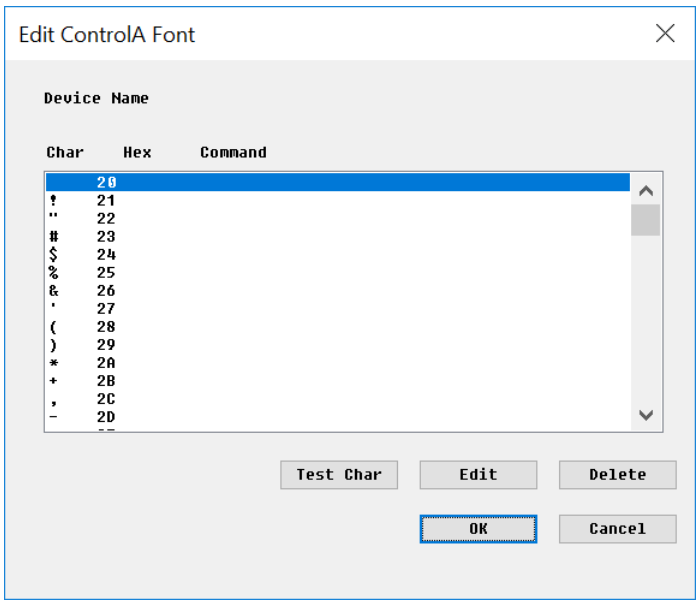
Click **Browse** then will show dialog to select the directory where log file is saved.

Log Setting Status	Data to be logged
Normal / Detail	Entry and Return, Receiving data and Error information when error happen
Normal	Sending data except printing data to printer
Detail	Sending data with Printing data

4.3.2 Edit ControlA

ControlA font is a font where specific command is set to control characters selected by the user. When the control character of the controlA font is specified in the print data and printed, the specific command is executed.

Driver convert character data to control code based on “Edit ControlA Font” Table then send command to printer.



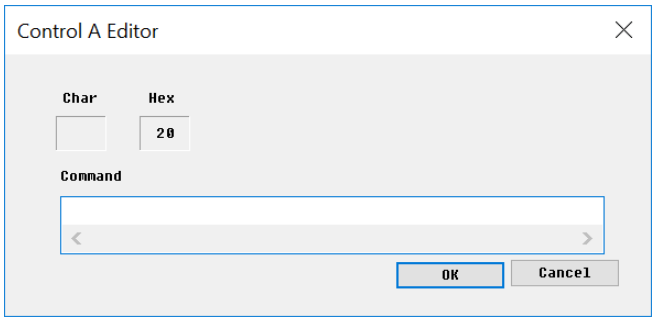
Test Char Button

To test the Command input for selected character

(Please see the example to use on Edit button section below)

Edit Button

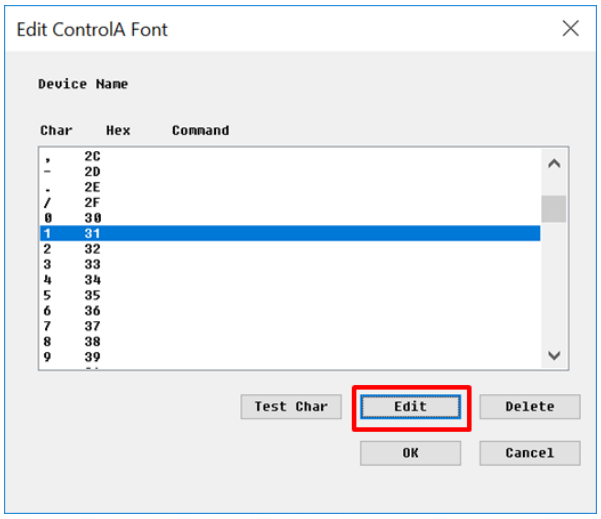
Show the dialog to edit Control A Font.



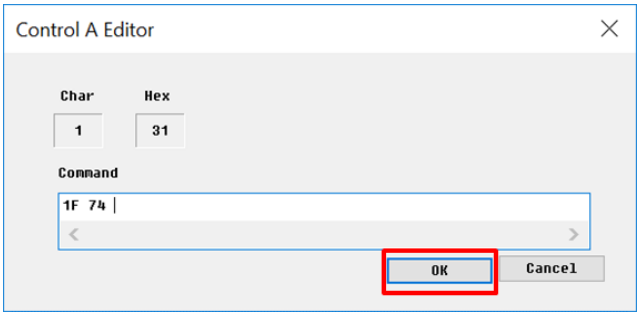
The command input will replace the selected character that is shown on "Char" section

Example:

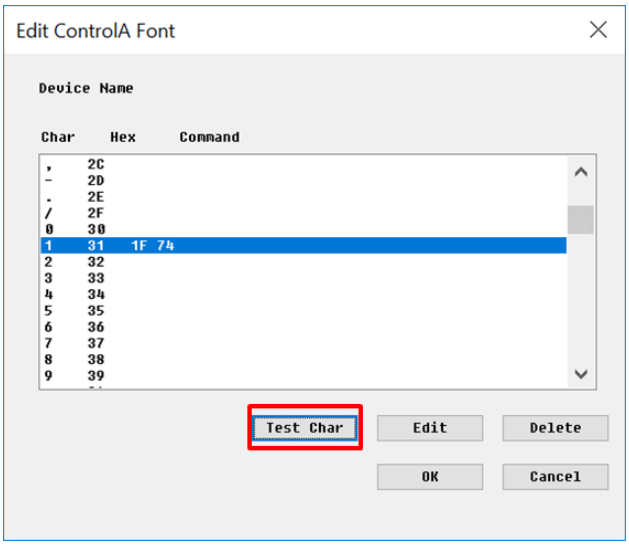
1. Select cursor to 1 Click Edit Button



2. Input Command then Click OK Button (Example : 1F 74)



3. Click Test Char Button



Result: Print Configuration Printing

Delete Button

Delete the command which will replace the selected character

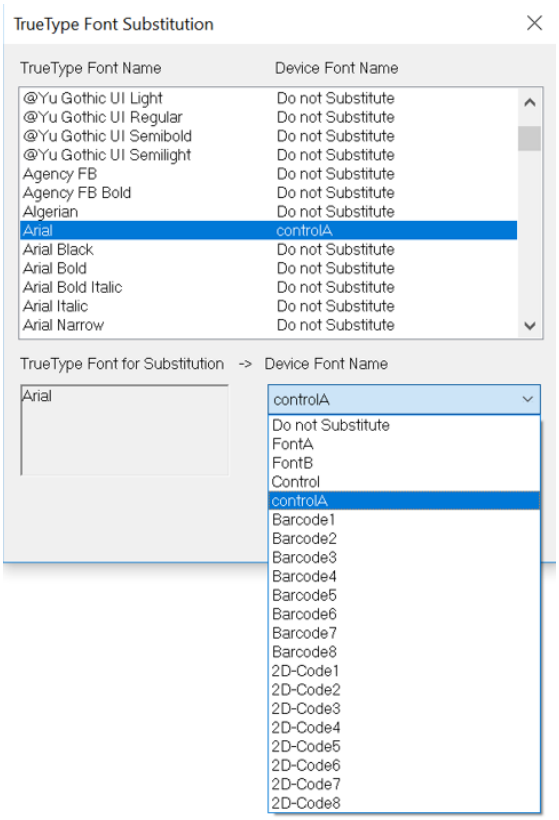
Note:

How to use the Edit ControlA in the document printing

1. Set command “1F 74” in Char “1” on Edit ControlA Font Setting
2. Open Printer properties → Printer Settings
3. Select True Type Font Substitution
4. Select Substitute option and click Advanced Settings

   Select True Type Font name and select Device Font Name to controlA

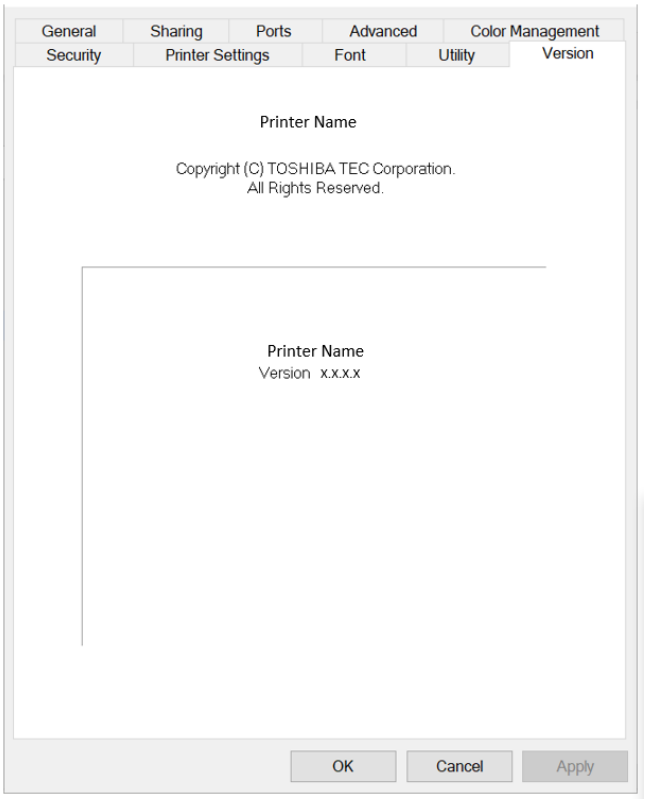
Example : Select Arial font



5. Click Apply & OK button
6. Open Microsoft Word document
7. Set font type to “Arial”
8. Type “1” then print dcument
9. Result is printing configuration page

4.4 Version

Show the version of printer driver.

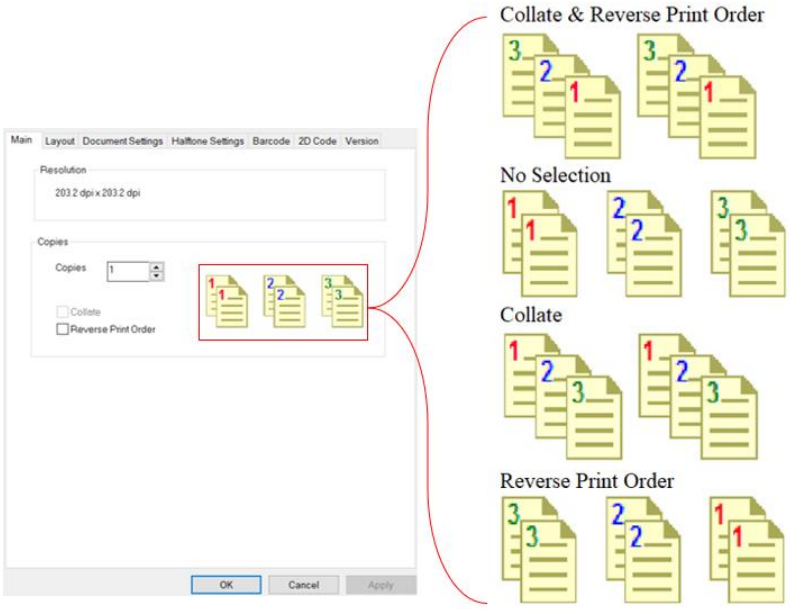


Note:

"Copyright (C) TOSHIBA TEC Corporation. All Rights Reserved" will not appear on TRST-P1X Printer, TRST-P2X Printer and TRST-L1X Printer.

5. Printing Preferences

5.1 Main



Copies

Input how many copies of every page will printed.

Collate

When printing more than one copy of a multi-page document, the copies will print all pages of each copy before printing the second copy.

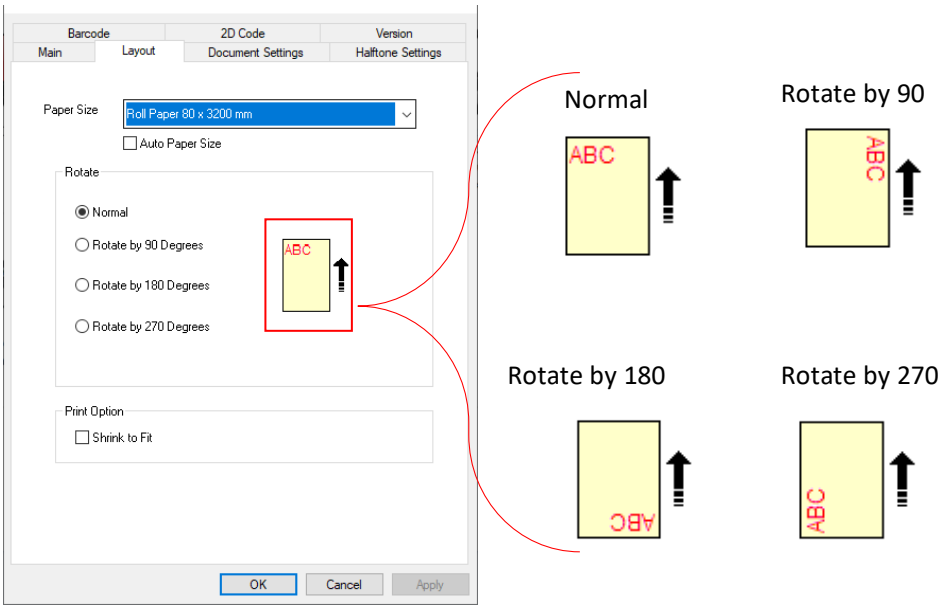
Reverse Print Order

Print order will start from the last page.

Note:

Partial cut support for TOSHIBATEC KOP-3X, TOSHIBATEC KOP-3S06 and TRST-L1X printer.

5.2 Layout



Paper Size

Select the specified paper size ([58mm x 297mm] or [80mm x 297mm]) based on paper width Selection:

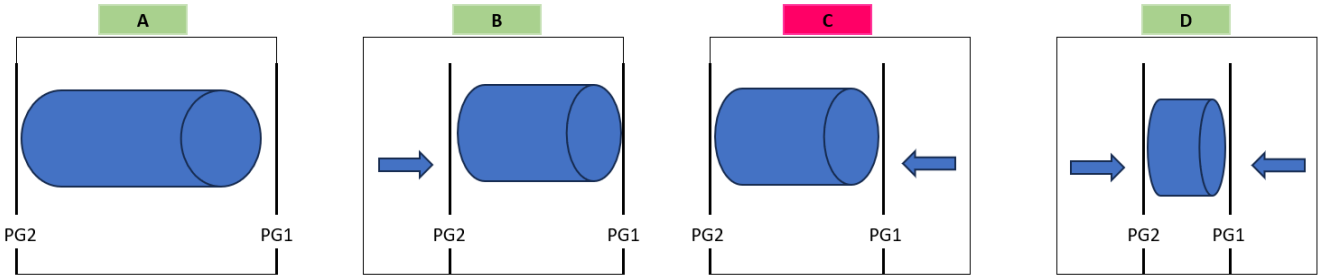
Paper Size	TRTS-P1X Printer	TRST-P2X Printer	TOSHIBATEC KOP-3X	TOSHIBA KOP-3S06	TRST-L1X Printer
Roll Paper 58 x 297 mm	O	O	O	O	O
Roll Paper 80 x 297 mm	O	O	O[1]	O[1]	O
Roll Paper 82 x 297 mm	x	x	O[2]	x	x
Roll Paper 58 x 3200 mm	O	O	O	O	O
Roll Paper 80 x 3200 mm	O[1]	O[1]	O	O	O[1]
Roll Paper 82 x 3200 mm	x	x	O[2]	x	x
Roll Paper 58 x 3200 mm without margin	O	O	O	O	O
Roll Paper 80 x 3200 mm without margin	O	O	O	O	O
Roll Paper 82 x 3200 mm without margin	x	x	O[2]	x	x
A4	O	O	x	x	x
User Defined Paper Size	O	O	O	O	O

[1] Default Paper Size

[2] Only support for KOP-3S01-A printer model. Default paper size will change into 82 mm x 297 mm for KOP-3S01-A model

Auto Paper Size

Auto paper size is to set automatically paper size based on paper guide position in printer. The position paper guide refers to below image:



*PG1 = Paper Guide 1
*PG2 = Paper Guide 2

A Position : 80 mm paper width use if both paper guides are in the home position. The driver will set the paper size to 80 x 3200 mm.

B Position : 58 mm paper width use if paper guide 1 in home position and paper guide 2 is set. The driver will set the paper size to 58 x 3200 mm.

C Position : Paper guide 1 is set and paper guide 2 in home position. This is an **illegal** set of paper guides, printer will set to busy and driver will set the paper size to 80 x 3200 mm.

D Position : 40 mm paper width use if both paper guides are set. Driver will set the paper size to 40 x 3200 mm.

Note:

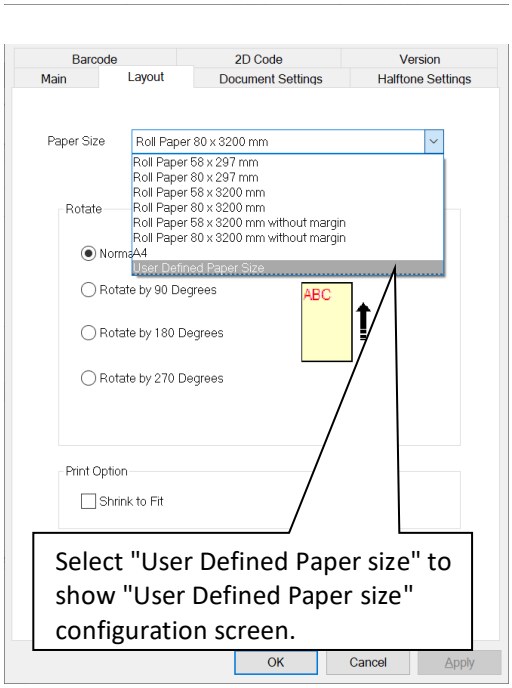
Auto Paper Size only support for TRST-L1X Printer.

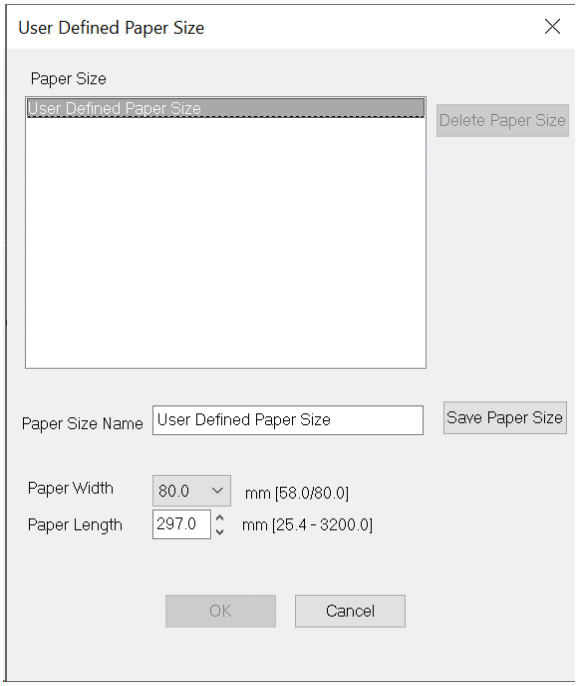
User Defined Paper Size

Custom the paper size by user.

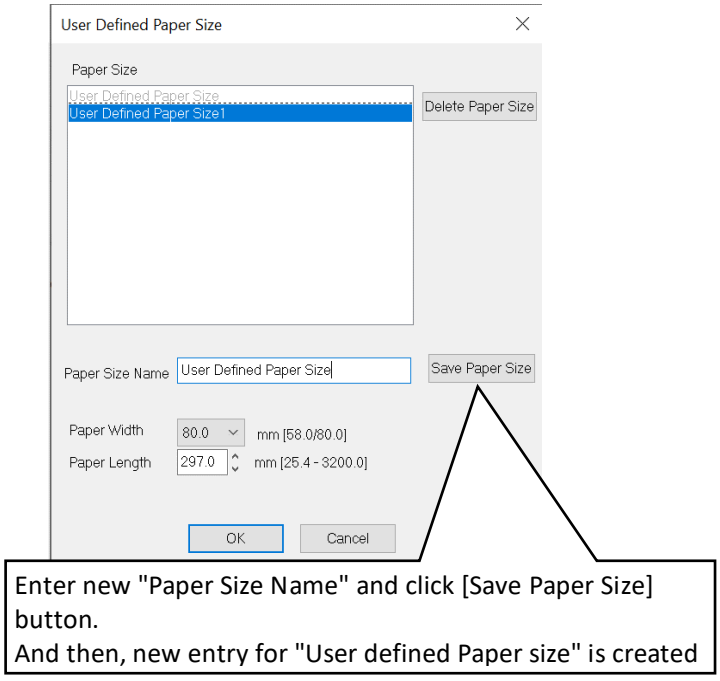
Example:

1. Call User Defined Paper size





2. Create new entry for "User defined Paper size"



3. Delete Paper Size

   To delete the selected "User Defined Paper Size"

Print Option

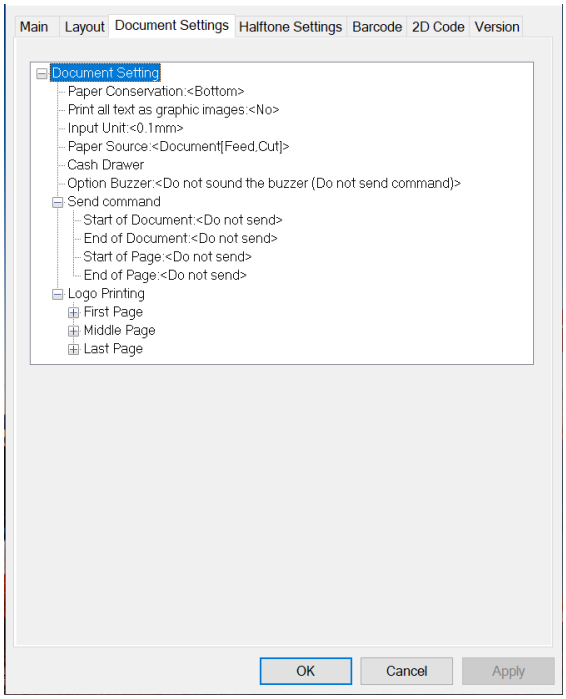
“Shrink to Fit” is Disable / Unselected (Default)

Printing result will cut off when paper width source is more than paper size setting.

“Shrink to Fit” is Enable / Selected

Printing result will scale down when paper width source is more than paper size setting.

5.3 Document Settings



Paper Conservation

Reduce paper usage by remove the empty area (Top / Bottom / Both area)

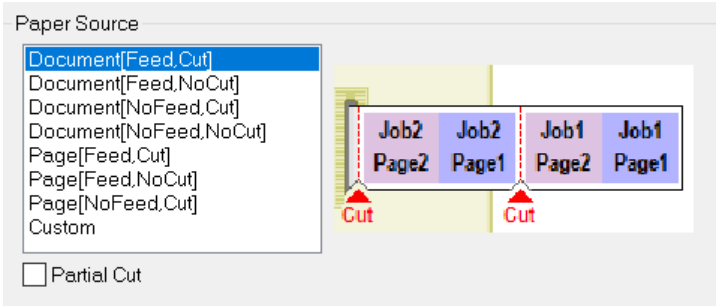
Print all text as graphic images

The printing data is printed as an image when “Yes” is selected

Input Unit

Set the input unit measurement parameter

Paper Source



Document[Feed, Cut]	Feed and cut on last page of a job
Document[Feed, No Cut]	Feed on last page of a job
Document[No Feed, Cut]	Cut on last page of a job
Document[No Feed, No Cut]	Neither Cut nor Feed are executed
Page[Feed, Cut]	Feed and cut between pages
Page[Feed, No Cut]	Feed between pages
Page[No Feed, Cut]	Cut between pages
Custom	Custom Setting

By default, the printer performs cuts based on the default cut settings supported by the printer. There are two types of cuts:

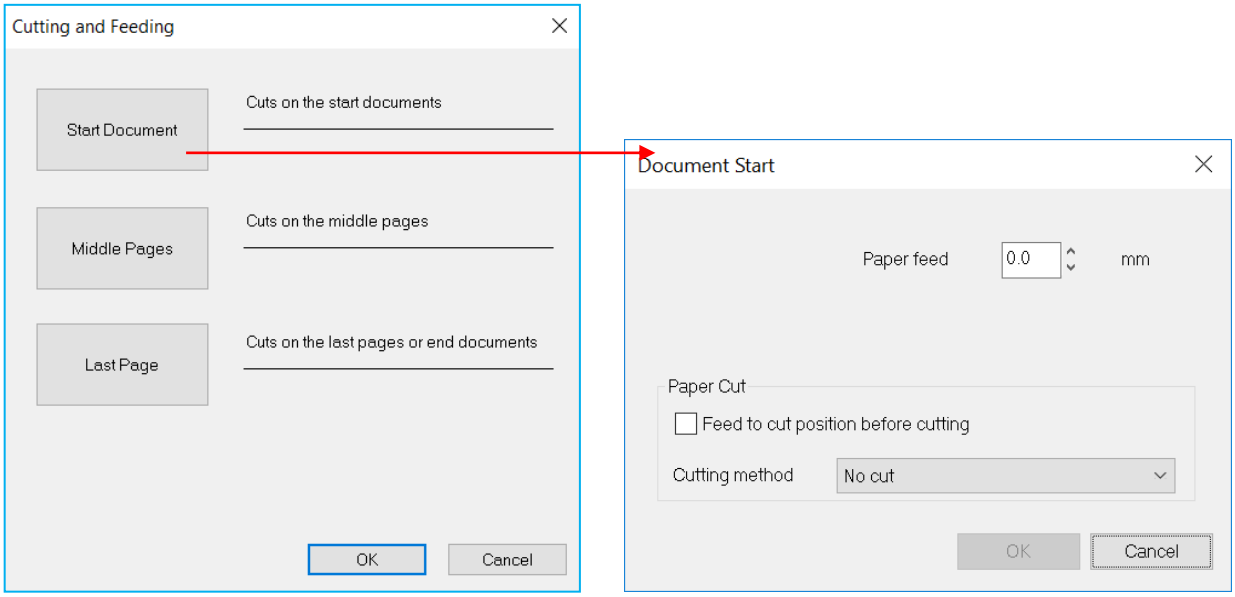
- Partial cut refers to cutting the paper in a way that leaves a small or uncut section.
- Full cut refers to completely cutting through the paper, fully separating it from the rest of the roll.

Some printers support both cut types, while others only support partial cuts. Please refer to the table below to see the supported printer models:

Printer Model	Cut supported	
	Partial cut	Full cut
TRST-P1X	✓	×
TRST-P2N	✓	×
TRST-P2N2	✓	✓
TRST-L1X	✓	✓
KOP-3X	✓	✓
KOP-306	✓	✓

For printers that do not support full cut, the Partial Cut checkbox will be hidden, and the printer will follow the default cut settings.

When Custom is selected then will show dialog as below



Start Document

Feed and cut controls on start of document

Middle Pages

Feed and cut controls on all pages except the last page when printing multiple pages. This function will be ignored if only a single page is printed

Last Page

Feed and cut controls on last page

Paper Feed

amount of Feed can be set between from 0mm to 100mm

Feed to cut position before cutting

If checked, feed with “Paper Feed” setting

Cutting method

Printer / Cut method	TRST-P1X / TRST-P2N	KOP-3X	KOP-3S06	TRST-L1X	TRST-P2N2
No cut	✓	✓	✓	✓	✓
Cut	✓	×	×	×	×
Full Cut	×	✓	✓	✓	✓
Partial Cut	×	✓	✓	✓	✓

Option Buzzer

Select buzzer sound in specific conditions:

Do not sound the buzzer (Do not send command)	Buzzer off or not sound the buzzer
Sound on at Start of Document	Buzzer will sound on start of document
Sound on at End of Document	Buzzer will sound on end of document
Sound on at Start of Page	Buzzer will sound on every start of page
Sound on at End of Page	Buzzer will sound on every end of page

Note:

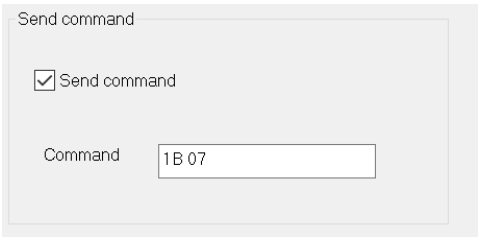
TRST-P1X Printer and TRST-L1X Printer are supported for option buzzer.

Send Command

Send Command of Start of Document, End of Document, Start of Page, and each End of Page

Example:

1. Input Start of Document : 1B 07 then click OK

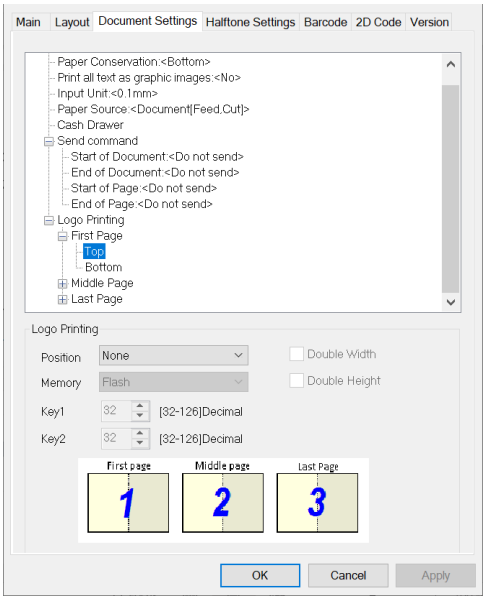


2. Do Print Test Page
3. Result is Beep Sound before print the test page

Logo Printing

The Logo print setting can be set to Top/Bottom of Last Page, Top/Bottom of First Page and Top/Bottom of Middle Page. Logo Position also can be set on Left, Center or Right.

The logo has to be downloaded before using this setting.



Position : Logo image is printed on Specified page/position (Left, Center or Right)

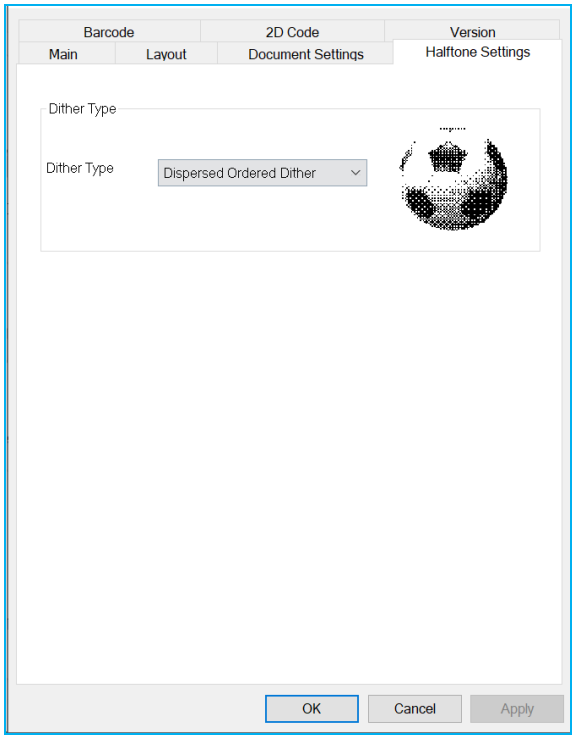
Key1 & Key 2 : It prints the user-defined logo by the key codes (kc1 and kc2) in flash memory

Double Width : Enlarge the logo image horizontally

Double Height : Enlarge the logo image vertically

5.4 Halftone Settings

Halftone is an image comprised of discrete dots rather than continuous tones.



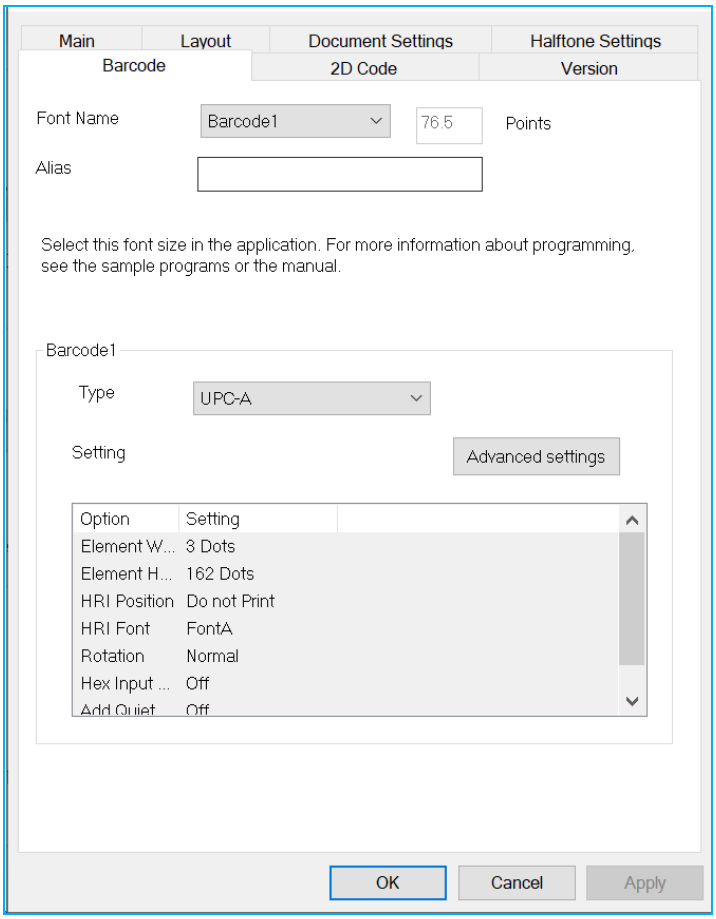
Dither Type setting to rendering the image:

Dither Type	Sample Image
Dispersed Ordered Dither	
Clustered Ordered Dither	
Error Diffusion	
Threshold	
Grayscale	

Note:

TOSHIBATEC KOP-3X printer not support for grayscale dither type.

5.5 Barcode



Font Name : There is 8 Barcode Font Name selection available.

Type:

Barcode1: UPC-A

Barcode2: UPC-E

Barcode3: JAN13

Barcode4: Code39

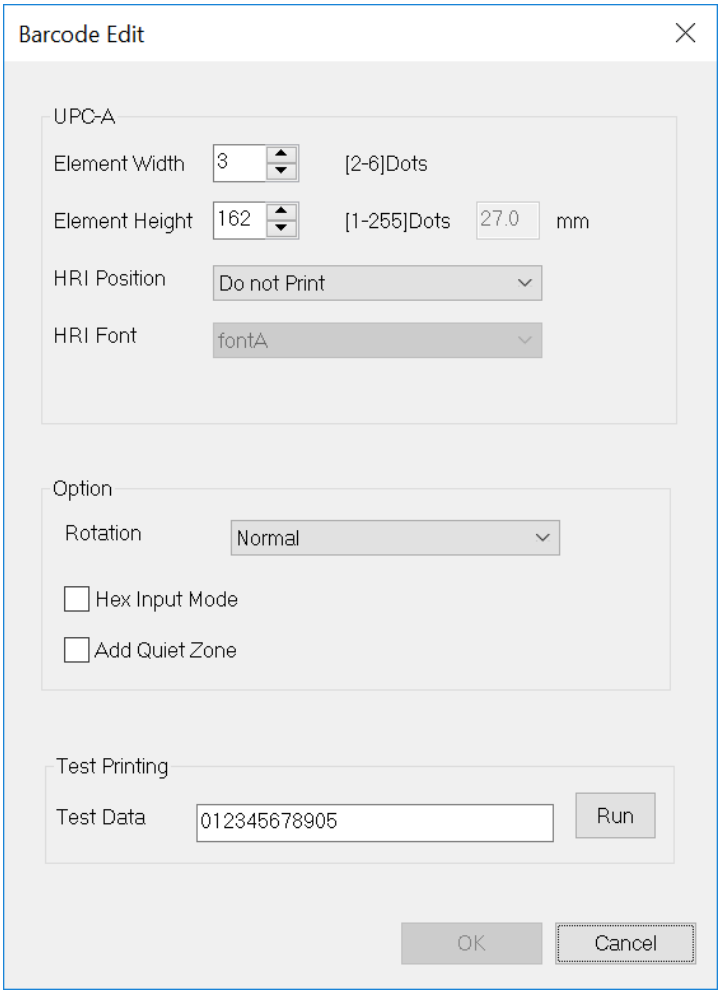
Barcode5: ITF

Barcode6: Codabar

Barcode7: Code93

Barcode8: Code128

Advanced Settings



Element Width : Barcode Width size in dots

Element Height : Barcode Height size in dots

HRI Position :

Human Readable Interpretation refers to the characters printed below, above or both side of Barcode Print

Rotation: Barcode rotation printing option

Hex input Mode :

This is the mode which accepts HEX data input from an application or Test Printing.

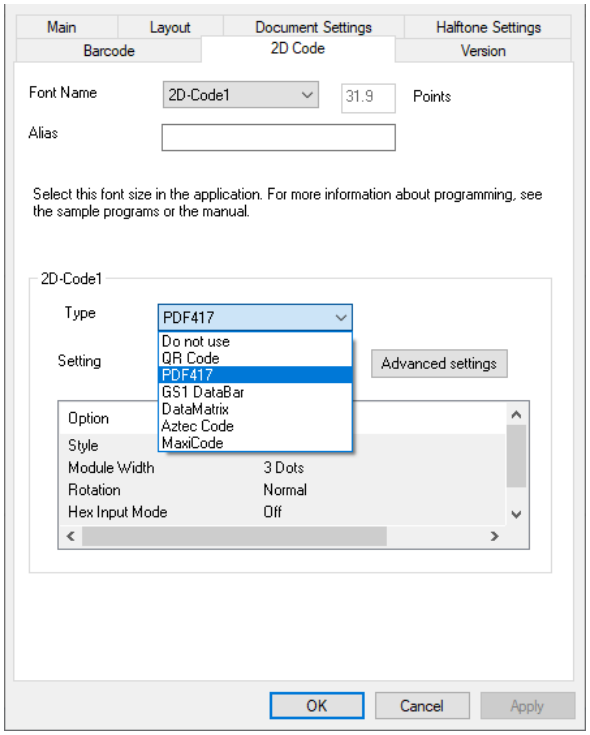
Example: In case to input "ABCJ" in Code-39 please type “4142434A” (Both ‘A’ and ’a’ is acceptable)

Add Quiet Zone :

Add the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode's symbology starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned

Test Data : Input the value to test the Barcode Print.

5.6 2D Code

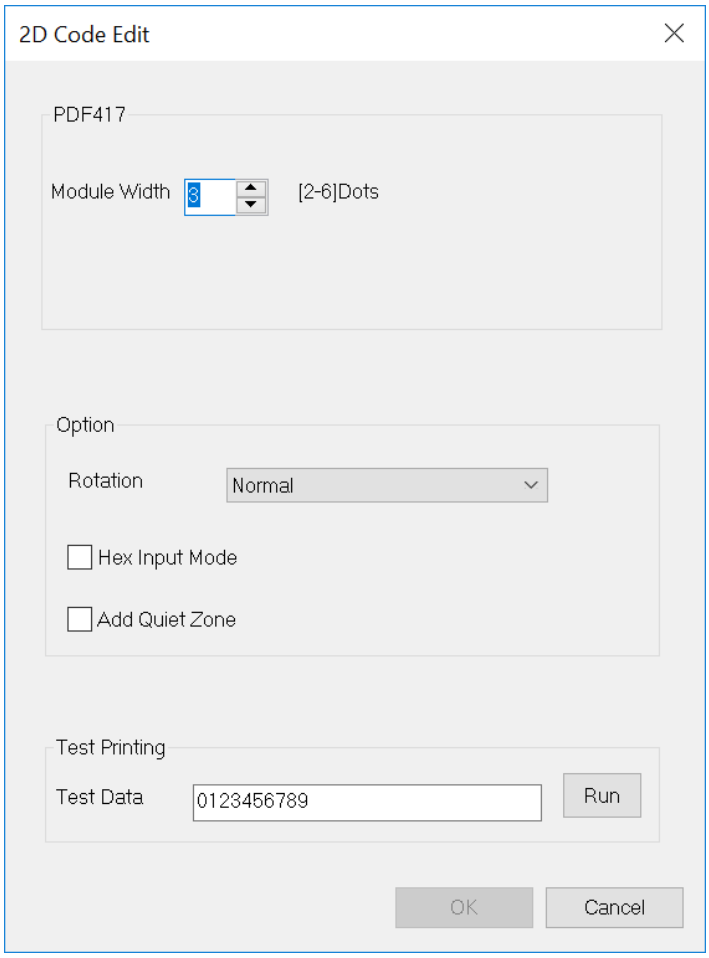


There are 8 predefined 2-D barcode fonts. To select/customize this, the user may select one of the font in “Font Name” drop box. “Alias” can be used to denote another name for each barcode font. The user is able to view/modify the type of barcode by selecting one of the option in “Type” drop down box. To view/modify more options, the user can click “Advanced Settings”.

**NOTE: To enable this feature, font substitution must be made from one of the TrueType Font to one of the 2-D barcode font. The substituted TrueType font must be used when wanting to use this feature.

5.6.1 PDF417

Advanced Settings



Module Width : Set the PDF417 Barcode Width size (in dots)

Rotation: PDF417 Barcode rotation printing option

Hex input Mode :

This is the mode which accepts HEX data input from an application or Test Printing.

Example: In case to input " 0123456789" please type "30313233343536373839"

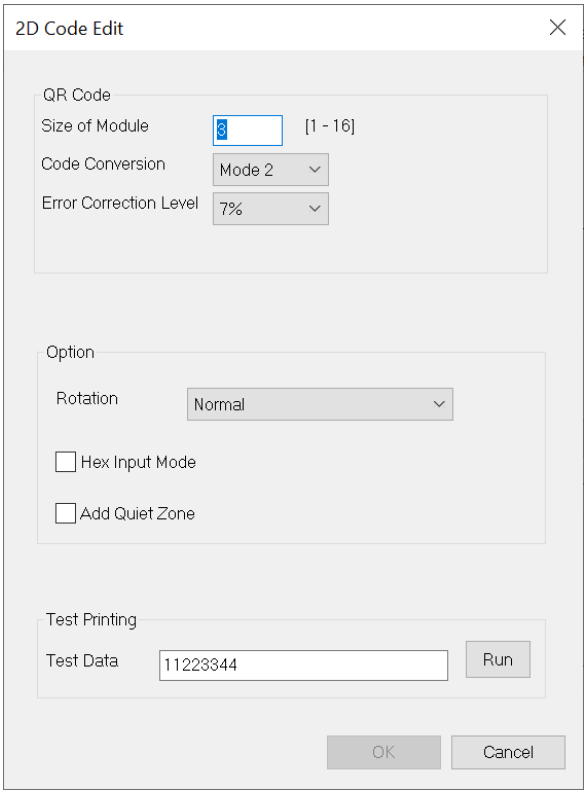
Add Quiet Zone :

Add the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode's symbology starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned

Test Data : Input the value to test the PDF417 Barcode Print.

5.6.2 QR Code

Advanced Settings



Size Module
User can set the QR code size in dots.

Code Conversion
User can select the model for QR code. The model are Model 1, Model 2 and Micro QR

Error Correction Level
User can select the Error Correction Level for QR code.

Rotation
QR Code rotation printing option

Hex Input Mode
Option for the QR Code data to be given as HEX data

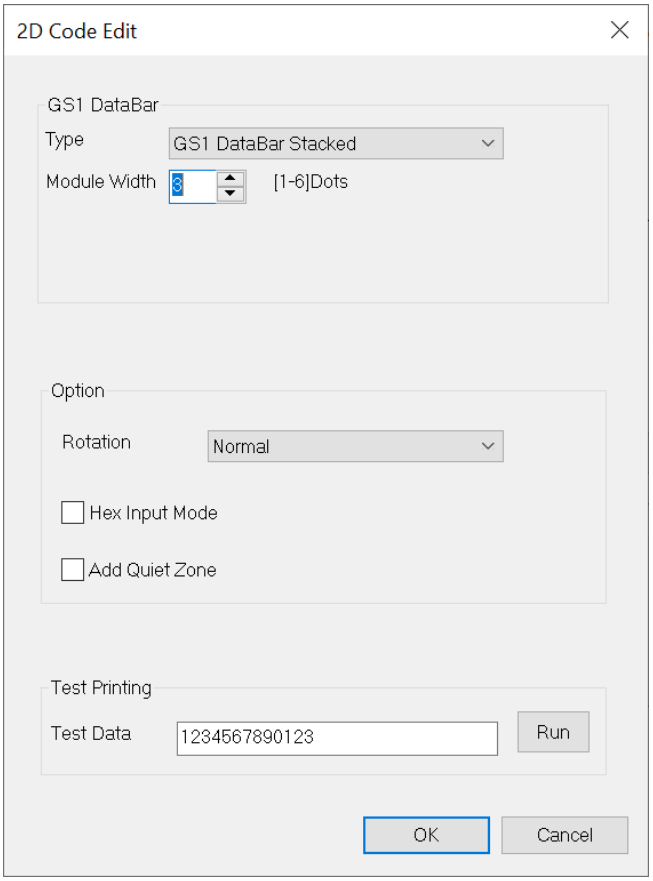
Add Quiet Zone
Option to add blank margins on either side of the QR Code to ensure the barcode reader is able to know where the barcode starts and stops. This is to ensure unintended data is not read by the reader

Test Data
“Test Data” text box can be filled with a sample data and pressing “Run” button will print out the QR Code with all the settings set by the user. Pressing the “OK” button will apply all settings to QR Code.

Limitation
Test Data maximum input is 128 character

5.6.3 GS1 DataBar

Advanced Settings



Type : Select GS1 DataBar Type.

The selection are GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional & GS1 DataBar Expanded Stacked.

Module Width : Set the GS1 DataBar Module Width.

Rotation : GS1 DataBar rotation printing option.

Hex input Mode : Option for the barcode data to be given as HEX data.

Add Quiet Zone :

Add the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode's symbology starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned

Test Data : Input the value to test the GS1 DataBar Print.

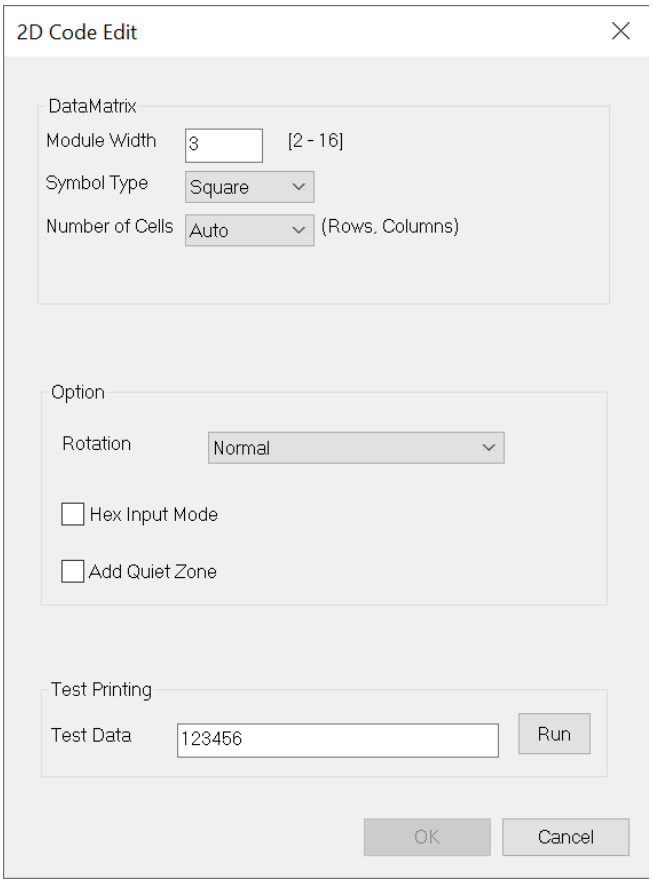
Note:

TOSHIBATEC KOP-3X not support GS1 DataBar.

Module width support for TRST-L1X are from 2 to 8.

5.6.4 DataMatrix

Advanced Settings



Module Width : Set the DataMatrix Module Width.

Symbol Type : Select DataMatrix Symbol Type.

The selection are Square and Rectangle.

Number of Cells : Select Rows and Column of DataMatrix based on Symbol Type selection.

Rotation : DataMatrix rotation printing option.

Hex input Mode : Option for the barcode data to be given as HEX data.

Add Quiet Zone :

Add the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode's symbology starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned

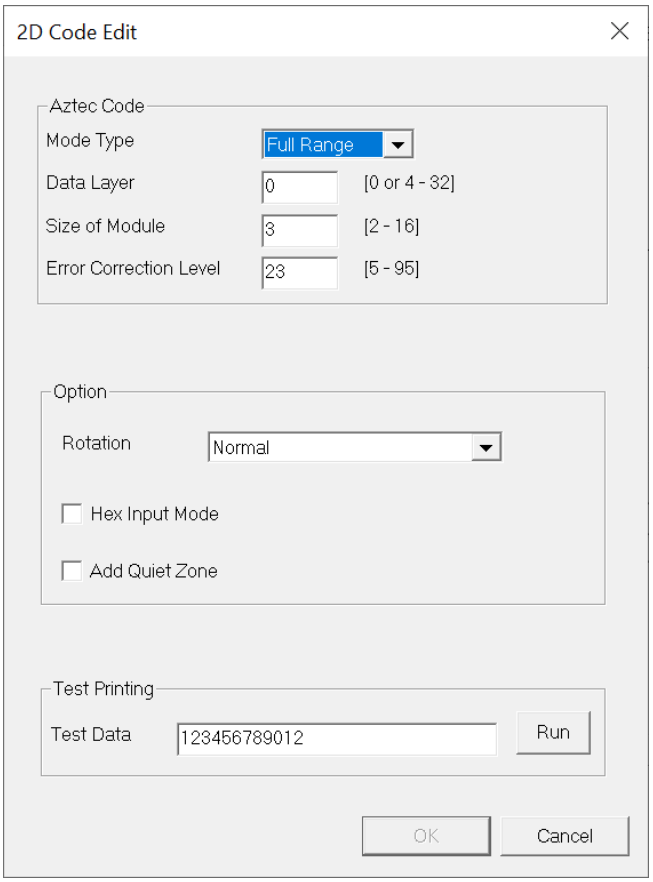
Test Data : Input the value to test the DataMatrix Print.

Note:

DataMatrix support only in TRST-L1X Printer.

5.6.5 Aztec Code

Advanced Settings



Mode Type : Select Aztec Code Mode.

The selection are Full Range and Compact.

Data Layer : Set number of data layers.

Size of Module : Set Aztec Code the code size in dots.

Rotation : Aztec Code rotation printing option.

Hex input Mode : Option for the barcode data to be given as HEX data.

Add Quiet Zone :

Add the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode's symbology starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned

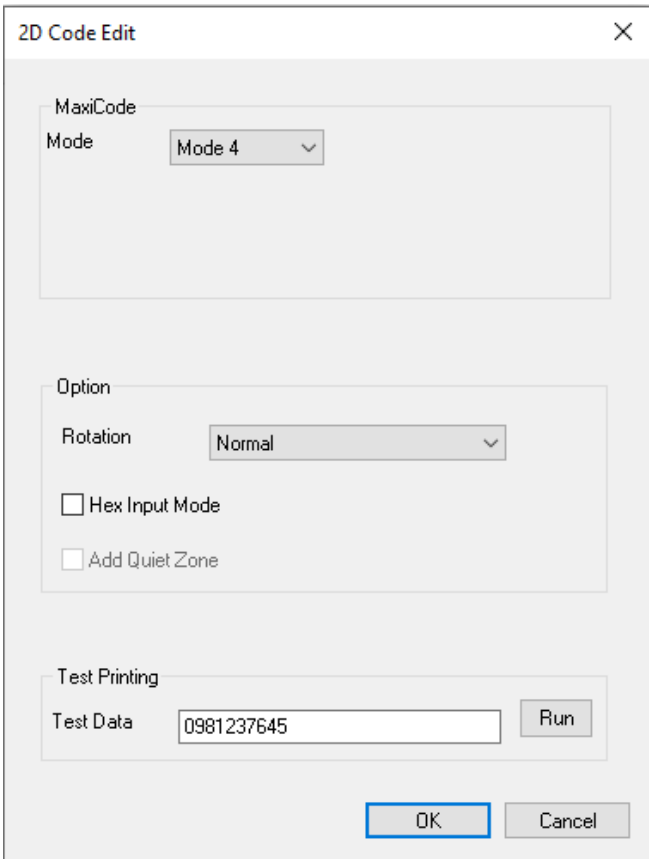
Test Data : Input the value to test the Aztec Code Print.

Note:

Aztec Code support only in TRST-L1X Printer.

5.6.6 MaxiCode

Advanced Settings



Mode: Select MaxiCode Mode.

The selection are Mode 2, Mode 3, Mode 4, Mode 5 and Mode 6.

Rotation : MaxiCode rotation printing option.

Hex input Mode : Option for the barcode data to be given as HEX data.

Mode 2 and Mode 3 must be use Hex as input data.

Add Quiet Zone :

Add the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode's symbology starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned

Test Data : Input the value to test the MaxiCode Print.

There are some requirements for Mode 2 and Mode 3 input data:

When mode 2 is selected the Primary Message includes all data except the following:

Factor of Primary Message	Number of bytes	Normal data
Postal code	1-9 byte	Numeric
ISO country code	1-3 byte	Numeric
Class of service code	1-3 byte	Numeric

When mode 3 is selected the Primary Message includes all data except the following:

Factor of Primary Message	Number of bytes	Normal data
Postal code	1-6 byte	Code set A
ISO country code	1-3 byte	Numeric
Class of service code	1-3 byte	Numeric

When using Mode 2 or 3, execute the process as listed below:
(**RS**, **GS** indicates control code of MaxiCode (**RS** = 1Eh, **GS** = 1Dh), "***yy***" indicates numeric data of 2 byte.)

a. In Primary Message, **GS** separates the message into Postal code, ISO country code and Class of service.
b. When the top of symbol data is "[)>" **RS** "01" **GS** "***yy***", these 9 bytes of data are treated as header.
   - Next data of the header is treated as Primary Message, then secondary Message
   - The format will be "[)>" **RS** "01" **GS** "***yy***"<Postal> **GS**<Country> **GS** <Service> **GS**<Secondary data>
c. When the top of symbol data is not "[)>" **RS** "01" **GS** "***yy***", the data is treated as Primary Message and remaining is treated as secondary. The format will be <Postal> **GS**<Country> **GS** <Service> **GS**<Secondary data>

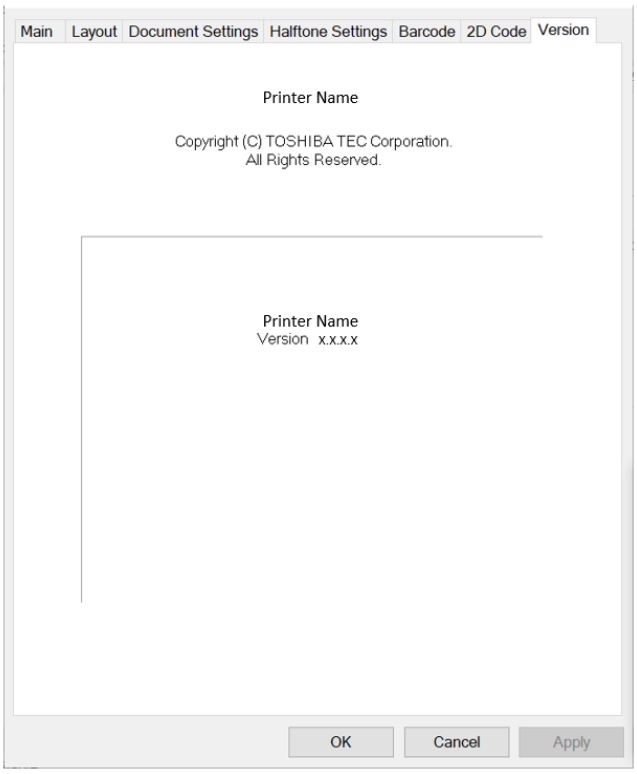
In mode 4,5 and 6, all the data in the symbol storage area is treated as primary message and secondary message. It does not check each code.

Note:

MaxiCode support only in TRST-L1X Printer.

5.7 Version

Show the version of printer driver



Note:

"Copyright (C) TOSHIBA TEC Corporation. All Rights Reserved" will not appear on TRST-P1X Printer, TRST-P2X Printer and TRST-L1X.

6. Printer Device Font

6.1 Introduction

This document contains sample snippets to explain on how to print device font on TRST-P1X, TRST-P2X, TOSHIBATEC KOP-3X, TOSHIBATEC KOP-3S06 and TRST-L1X Printer Thermal Receipt printers with Windows Printer Driver

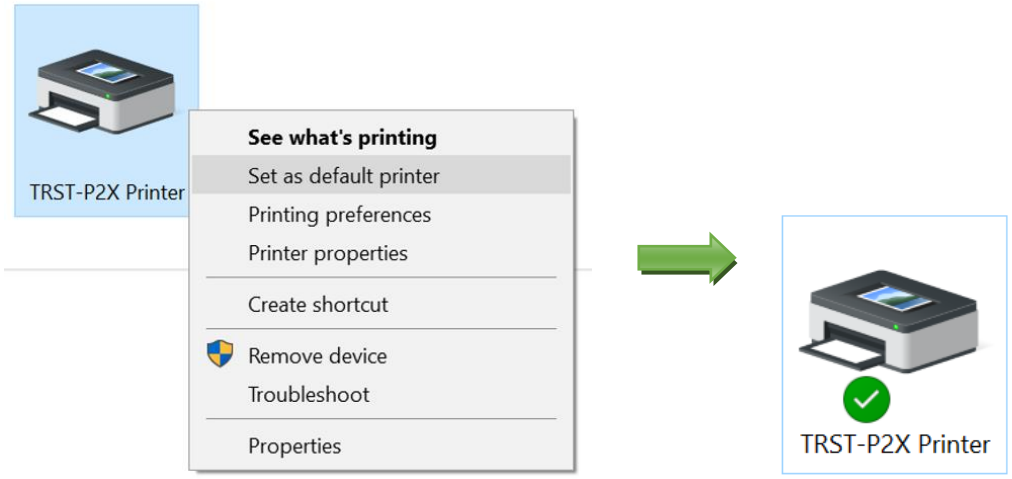
6.2 Setup Printer as Default Printer

1. Choose Start → Devices and Printers.

   The Devices and Printers window appears. The current default printer is indicated by a check mark.

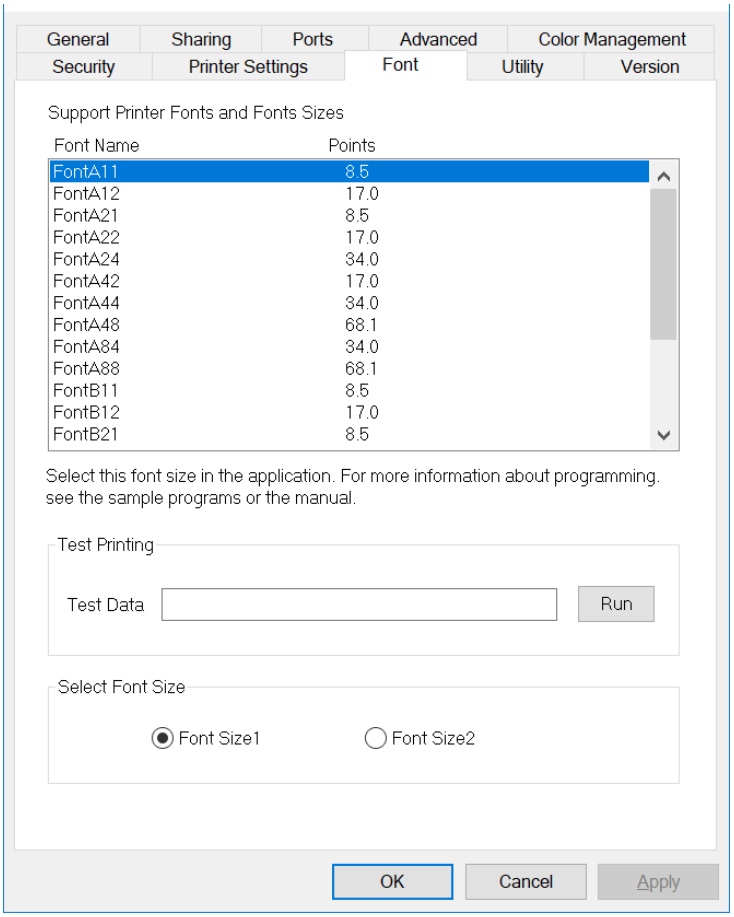
2. Right-click any printer that isn't set as the default and choose "Set as Default Printer" from the shortcut menu.

   If you right-click the printer that is already set as the default, the Set as Default Printer command won't be available on the shortcut menu.



6.3 Supported Fonts

The following font names are supported by the Windows printer driver models.



6.4 Sample Code

Note: Before you run the code, please install the Windows Printer Driver and make sure the printer is set as default printer.

6.4.1 Visual Basic 6.0

Printer Object

Visual Basic contains a global printer object, which refers to the default printer for the current system. Because this object is global to all parts of the VB project, you don't need to create an object variable, you can simply use the Printer object directly. Please refer *properties* and *methods* of the Printer Object in the following link.

Reference: please visit *https://docs.microsoft.com/en-us/previous-versions/bb882722(v%3Dvs.140)*

How To:

1. Create a New Standard EXE Project in Visual Basic 6.0.
2. Copy the following sample snippet.
3. Call the Function with an event (Ex. Button OnClick()).
4. The Printer will print the text.

```
Public Sub Print_FontA11()
    Printer.Font.Size = 11
    Printer.Font.Name = "FontA11"
    Printer.Print "FontA11"
    Printer.EndDoc
End Sub

Public Sub Print_FontA12()
    Printer.Font.Size = 19.9
    Printer.Font.Name = "FontA12"
    Printer.Print "FontA12"
    Printer.EndDoc
End Sub

Public Sub Print_FontA21()
    Printer.Font.Size = 9.9
    Printer.Font.Name = "FontA21"
    Printer.Print "FontA21"
    Printer.EndDoc
End Sub
```

6.4.2 Visual C++ (MFC)

6.4.2.1 Using Print Dialog

The following snippet prints a text with device font.

1. Create a variable with CPrintDialog with FALSE constructor.

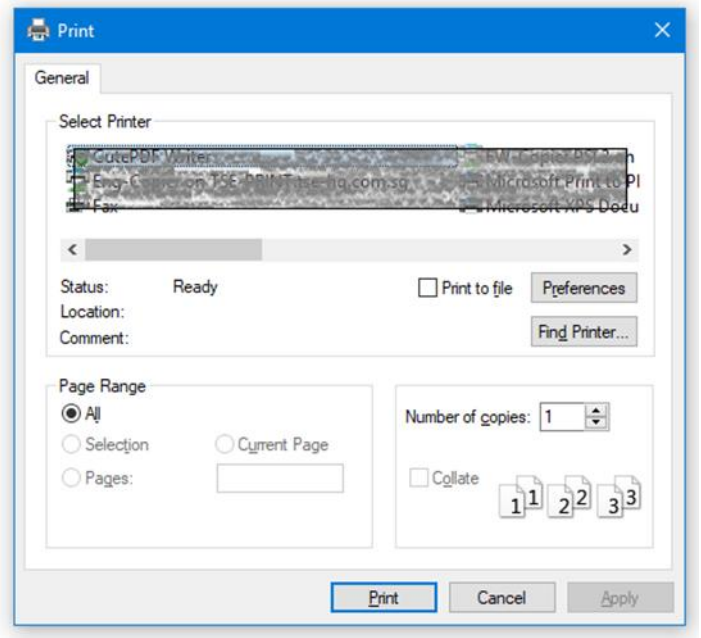
```
CPrintDialog( BOOL bPrintSetupOnly, DWORD dwFlags = PD_ALLPAGES |
PD_USEDEVMODECOPIES | PD_NOPAGENUMS | PD_HIDEPRINTTOFILE |
PD_NOSELECTION, CWnd* pParentWnd = NULL);

bPrintSetupOnly

Specifies whether the standard Windows Print dialog box or Print Setup
dialog box is displayed. Set this parameter to TRUE to display the standard
Windows Print Setup dialog box. Set it to FALSE to display the Windows
Print dialog box. If bPrintSetupOnly is FALSE, a Print Setup option button
is still displayed in the Print dialog box.
```

2. Get Printer device defaults without displaying a dialog box.

```
dlg.GetDefaults();
```

Note: If you want show the print dialog, call the DoModal() and get the settings.

```
CPrintDialog dlg(FALSE);

if (dlg.DoModal() == IDOK)

{

}
```



3. Get a handle to the printer device context (DC).

```
HDC hdcPrinter = dlg.GetPrinterDC();
```

4. Create the print document using DOCINFO structure.
5. Attach the handle Device Context “hdcPrinter” to “CDC dcPrinter”.
6. Create the device font using CreateFontIndirect() and select the font for the device context.

```
LOGFONTA lf;
strcpy (lf.lfFaceName, "FontA42"); // Device Font Name
HFONT hfont = CreateFontIndirect(&lf);
HFONT holdfont = (HFONT)SelectObject(dcPrinter, hfont);
```

7. Draw the Text or any shapes to the CDC.

```
dcPrinter.TextOut(10 ,10, _T("FontA42 Regular"));
```

Reference: *https://docs.microsoft.com/en-us/cpp/mfc/reference/cdc-class?view=vs-2019*

8. End the document page.

```
dcPrinter.EndPage();
dcPrinter.EndDoc();
```

9. Cleanup the font objects to release the memory.

Create a VC++ (MFC) Project and call the following function.

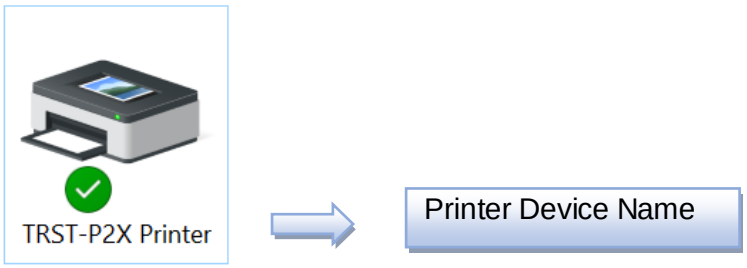
```
void PrintSampleFonts()
{
 // Declaration
 CPrintDialog dlg(FALSE);
 HDC hdcPrinter;
 CDC dcPrinter;
 DOCINFO docinfo;
 LOGFONTA lf;
 HFONT hfont;
 HFONT  holdfont;
 int _XPOS;
 // Initialization
 dlg.GetDefaults();
 hdcPrinter = dlg.GetPrinterDC();
 memset(&lf , 0x00 , sizeof(LOGFONTA));
 memset(&docinfo,0,sizeof(docinfo));
 XPOS = 10;
 if(hdcPrinter == NULL)
 {
    MessageBox("Printer Not Initialized ");
 }
 else
 {
    dcPrinter.Attach(hdcPrinter);
    docinfo.cbSize = sizeof(docinfo);
    docinfo.lpszDocName = _T("DeviceFontSample");
    if(dcPrinter.StartDoc(&docinfo) < 0)
    {
       MessageBox("StartDoc failed.");
    }
    else
    {
       // Start a page
       if(dcPrinter.StartPage() < 0 )
       {
          MessageBox("Could not start page");
          dcPrinter.AbortDoc();
       }
       else
       {
          // Font A42 Regular
          strcpy (lf.lfFaceName, "FontA42");
          hfont = CreateFontIndirect(&lf);
          holdfont = (HFONT)SelectObject(dcPrinter, hfont);
          dcPrinter.TextOut(_XPOS ,10, _T("FontA42 Regular"));
          // Font A21 Regular
          memset(&lf , 0x00 , sizeof(LOGFONTA));
          strcpy (lf.lfFaceName, "FontA21");
          hfont = CreateFontIndirect(&lf);
          holdfont = (HFONT)SelectObject(dcPrinter, hfont);
          dcPrinter.TextOut(_XPOS,40, _T("FontA21 Regular"));
          // Font A22 Regular
          memset(&lf , 0x00 , sizeof(LOGFONTA));
          strcpy (lf.lfFaceName, "FontA22");
          hfont = CreateFontIndirect(&lf);
          holdfont = (HFONT)SelectObject(dcPrinter, hfont);
          dcPrinter.TextOut(_XPOS,80, _T("FontA22 Regular"));

          dcPrinter.EndPage();
          cPrinter.EndDoc();

          // Clean up.
          SelectObject(dcPrinter, holdfont);
          DeleteObject(hfont);
       }
    }
 }
}
```

This is the device font Name provided by driver.

6.4.2.2 Direct Call using Printer Name

1. Create a variable for HDC and Create Device Context using CreateDC().constructor. Enter the printer name as highlighted below

```
HDC hdc = NULL;
hdc = CreateDC(TEXT("WINSPOOL"), TEXT("TRST-P2X Printer"), NULL , NULL);
```



The following sample has only difference from above code is calling the printer without print dialog.

Create a VC++ (MFC) Project and call the following function.

```
void PrintSampleFonts()
{
	// Declaration
	HDC hdc = NULL;
	hdc = CreateDC(TEXT("WINSPOOL"), TEXT("TRST-P2X Printer"), NULL , NULL);
	HDC hdcPrinter;
	CDC dcPrinter;
	DOCINFO docinfo;
	LOGFONTA lf;
	HFONT hfont;
	HFONT  holdfont;
	int _XPOS;
	// Initialization
	dlg.GetDefaults();
	hdcPrinter = dlg.GetPrinterDC();
	memset(&lf , 0x00 , sizeof(LOGFONTA));
	memset(&docinfo,0,sizeof(docinfo));
	_XPOS = 10;
	if(hdcPrinter == NULL)
	{
		MessageBox("Printer Not Initialized ");
	}
	else
	{
		dcPrinter.Attach(hdcPrinter);
		docinfo.cbSize = sizeof(docinfo);
		docinfo.lpszDocName  = _T("DeviceFontSample");
		if(dcPrinter.StartDoc(&docinfo) < 0)
		{
		    MessageBox("StartDoc failed.");
		}
		else   // Start a page
		{      if(dcPrinter.StartPage() < 0 )
		       {
		           MessageBox("Could not start page");
		           dcPrinter.AbortDoc();
		       }
		       else
		       {
		              // Font A42 Regular
		              strcpy (lf.lfFaceName, "FontA42");
		              hfont = CreateFontIndirect(&lf);
		              holdfont = (HFONT)SelectObject(dcPrinter, hfont);
		              dcPrinter.TextOut(_XPOS ,10, _T("FontA42 Regular"));

		              // Font A21 Regular
		              memset(&lf , 0x00 , sizeof(LOGFONTA));
		              strcpy (lf.lfFaceName, "FontA21");
		              hfont = CreateFontIndirect(&lf);
		              holdfont = (HFONT)SelectObject(dcPrinter, hfont);
		              dcPrinter.TextOut(_XPOS,40, _T("FontA21 Regular"));

		              // Font A22 Bold Regular
		              memset(&lf , 0x00 , sizeof(LOGFONTA));
		              strcpy (lf.lfFaceName, "FontA22");
		              hfont = CreateFontIndirect(&lf);
		              holdfont = (HFONT)SelectObject(dcPrinter, hfont);
		              dcPrinter.TextOut(_XPOS,80, _T("FontA22 Regular"));


		              dcPrinter.EndPage();
		              dcPrinter.EndDoc();
		              // Clean up.
		              SelectObject(dcPrinter, holdfont);
		              DeleteObject(hfont);
		        }
		}
	}
```

6.5 References

1. MSDN - *https://support.microsoft.com/en-us/help/201978/how-to-use-printer-device-fonts*
2. *https://www.oreilly.com/library/view/vb-vba/1565923588/1565923588_ch07-1723-fm2xml.html*
3. CPrintDialog - *https://docs.microsoft.com/en-us/cpp/mfc/reference/cprintdialog-class?view=vs-2019*
4. CreateFontIndirectA - *https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-createfontindirecta*
5. LOGFONTA structure - *https://msdn.microsoft.com/en-us/ie/aa741230(v=vs.94)*

There are two ***.iss** file that needed to run silent installation and uninstallation.

One is for installation (**install.iss**) , and the other is for uninstallation (**uninstall.iss**).

Both files are generated using executable installer.

User need to generate one time before using it for silent mode installation.

Windows installation / uninstallation dialog will be prompted when creating *.iss file.

7. Creating *.iss file for driver installation (silent mode)

There are two ***.iss** file that needed to run silent installation and uninstallation.

One is for installation (**install.iss**) , and the other is for uninstallation (**uninstall.iss**).

Both files are generated using executable installer.

User need to generate one time before using it for silent mode installation.

Windows installation / uninstallation dialog will be prompted when creating *.iss file.

7.1 Steps to generate install.iss

1. Run below script via command line (path must be same with executable installer location):

   [Setup.exe] /r /f1**[install.iss destination path]**

   Example:

   [Setup.exe] → TEC_WinDriver_Vx.x.x.x.exe

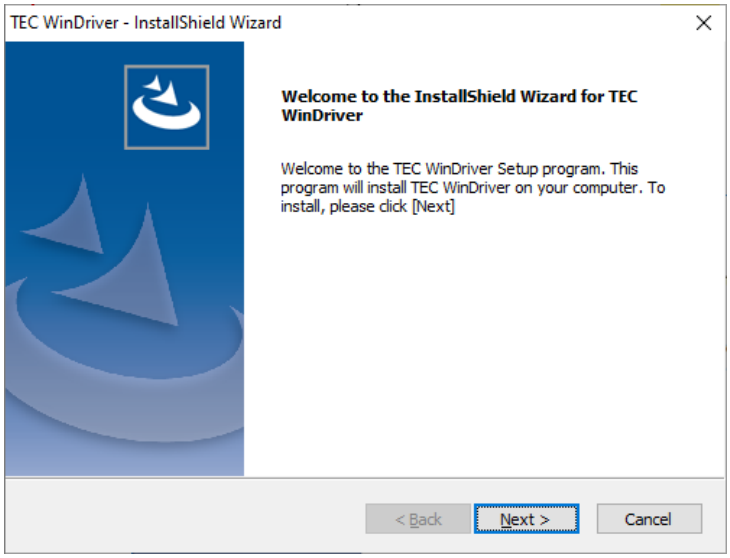
   [install.iss destination path] → C:/temp/install.iss

   Script will be:

   TEC_WinDriver_Vx.x.x.x.exe /r /f1**C:/temp/install.iss**

2. Windows installation dialog prompt after step 1.

   Please complete the installation steps (like windows manual installation steps).



3. **install.iss** file will be generated in destination path

7.2 Steps to generate uninstall.iss

Note: Must do *steps to generate **install.iss*** first.

1. Run below script via command line (path must be same with executable installer location):

   [Setup.exe] /r /f1**[uninstall.iss destination path]**

   Example:

   [Setup.exe] → DieboldNixdorf_WindowsDriver_Vxx.xx.xx.xx.exe

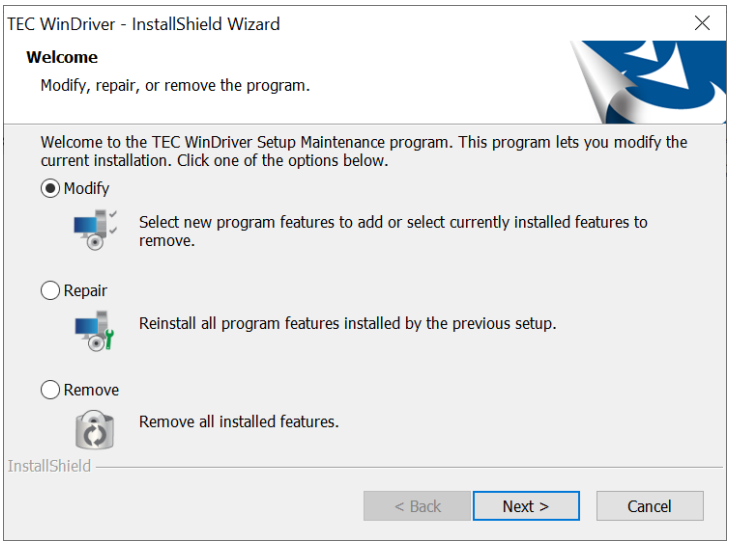
   [uninstall.iss destination path] → C:/temp/uninstall.iss

   Script will be:

   DieboldNixdorf_WindowsDriver_Vxx.xx.xx.xx.exe /r /f1**C:/temp/uninstall.iss**

2. Windows uninstallation dialog prompt after step 1.

   Please complete uninstallation steps (like manual windows uninstallation steps).



3. **uninstall.iss** file will be generated in destination path

8. API

8.1 Introduction

This section describes the API (application programming interface) which is the layer between Application and Printer. This section will also detail out the expected input and output of each function and its behavior.

8.2 Supported Printer

Windows Common Printer Driver API supports the following printers.

- TRST-P1X
- TRST-P2X
- KOP-3X
- KOP-3S06
- TRST-L1X

8.3 System Requirement

<OS>
Microsoft® Windows™10
Microsoft® Windows™11

8.4 Supported API VS Printer

Below is table that describe supported API for each printer

Printer / API	TRST-P1X / TRST-P2X	KOP-3X	KOP-3S06	TRST-L1X
TPPIOpen	✓	✓	✓	✓
TPPIClose	✓	✓	✓	✓
TPPIGetStatus	✓	✓	✓	✓
TPPITxRx	✓	✓	✓	✓
TPPICtrlTxRx	✓	✓	✓	✓
TPPIPartialCut_Declare	×	✓	×	×
TPPIPartialCut_Perform	×	✓	×	×
TPPIPartialCut_FeedCut	×	✓	✓	×
TPPIRetract_Eject	×	✓	×	×
TPPIRetract_Retract	×	✓	×	×
TPPIRetract_RetryCount	×	✓	×	×
TPPIRetract_Timeout	×	✓	×	×
TPPIBezel_Opt	×	✓	✓	✓
TPPIBezel_AutoMode	×	✓	✓	✓
TPPIBezel_Usermode	×	✓	✓	✓
TPPIBezel_Configure	×	✓	✓	✓

✓ Indicates the API is supported on related Printer Model

× Indicates the API is not supported on related Printer Model

8.5 API Module

8.5.1 TPPIOpen

Function Name	TPPIOpen
Function Prototype	int TPPIOpen(IN LPWSTR lpstrPrinterName, INOUT HANDLE *hHandle)
Input	LPWSTR lpstrPrinterName Name of target printer to open the connection to HANDLE *hHandle Pointer to the handle
Output	HANDLE * hHandle 0 – function failed <>0 – function success
Description	This function opens a connection to the specified printer Printer status monitor starts
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_SPOOL_SUSPEND Spooler service is not available DRV_ERR_PRINTER_CANNOT_OPEN Printer cannot be opened DRV_ERR_CANNOT_SEND_DATA Driver unable to send data to printer DRV_ERR_CONNECTION_TIMEOUT Printer is not connected or power off No response from the printer after specific time DRV_ERR_INTERNAL Error exist between API and language monitor
Comments	Any status/error changes that occur before this function is called are not monitored. The handle that is returned by this function should be used for the handle for other API calls. “DRV_ERR_SPOOL_SUSPEND” is returned will also be returned after the third ‘open’ failure

8.5.2 TPPIClose

Function Name	TPPIClose
Function Prototype	int TPPIClose (IN HANDLE hHandle)
Input	HANDLE hHandle Handle received by ‘Open’ function for target printer
Output	None
Description	This function closes the connection to the printer linked to the handle Printer status monitor stops after function complete
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CONNECTION_NOT_OPENED TPPIOpen() was not called previously
Comments	The printing movement running on the printer is stopped. This function will still complete regardless of error.

8.5.3 TPPIGetStatus

Function Name	TPPIGetStatus
Function Prototype	int TPPIGetStatus (IN HANDLE hHandle, OUT STSDATA *STSDATA);
Input	HANDLE hHandle Handle received by ‘Open’ function for target printer
Output	STSDATA *STSData
Description	This function returns all of the printer’s status.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CONNECTION_TIMEOUT The printer is not connected or power off No response is returned from the printer in a set time DRV_ERR_CONNECTION_NOT_OPENED TPPIOpen() was not called previously DRV_ERR_INVALID_PARAMETER Input parameter is not correct DRV_ERR_INTERNAL Error occurred between the API and the language monitor
Comments	Refer to the corresponding Windows Printer Driver documentation for STSDATA data structure for its size and further details.

8.5.4 TPPITxRx

Function Name	TPPITxRx
Function Prototype	int TPPITxRx (IN HANDLE hHandle, IN DWORD dwTxLen, IN LPBYTE lpbTxBuff, INOUT LPDWORD lpdwRxLen, OUT LPBYTE lpbRxBuff, IN DWORD dwTimeout);
Input	HANDLE hHandle Handle received by 'Open' function for target printer DWORD dwTxLen The length of the data to be sent to the printer LPBYTE lpbTxBuff Data to be sent to the printer LPDWORD lpdwRxLen The expected length of the data to be received from the printer DWORD dwTimeout The length of the data received from the printer.
Output	LPDWORD lpdwRxLen The actual length of the data received from the printer LPBYTE lpbRxBuff Data received from the printer if lpdwRxLen is not NULL
Description	This function sends data to printer as a bulk command If a request was made, receives the data from the printer
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_RESPONSE Response from the printer is invalid. DRV_ERR_CONNECTION_TIMEOUT The printer is not connected or power off No response is returned from the printer in a set time DRV_ERR_CONNECTION_NOT_OPENED TPPIOpen() was not called previously DRV_ERR_INVALID_PARAMETER Input parameter is not correct DRV_ERR_PRINTING_ALREADY_IN_PROGRESS Print job is already sent to the printer by GDI DRV_ERR_INTERNAL Error occurred between the API and the language monitor
Comments	• If lpbTxBuff is NULL, function returns "DRV_ERR_INVALID_PARAMETER" • If dwTXLen is NULL, function returns "DRV_ERR_INVALID_PARAMETER" • If lpdwRxLen is not NULL and no reponse is received after exceeding dwTimeout, this function return with lpdwRxLen=0 and a specific error • TPPITxRx only sent the command without check the printer condition • If the length of expected reply is longer than the actual reply of the command, this API will wait until it timed out before returning. Function returns "DRV_ERR_INVALID_RESPONSE" • If the length of expected reply is shorter than the actual reply of the command, it will return "DRV_ERR_INVALID_RESPONSE" immediately

8.5.5 TPPICtrlTxRx

Function Name	TPPICtrlTxRx
Function Prototype	int TPPICtrlTxRx (IN HANDLE hHandle, IN DWORD dwTxLen, IN LPBYTE lpbTxBuff, INOUT LPDWORD lpdwRxLen, OUT LPBYTE lpbRxBuff, IN DWORD dwTimeout);
Input	HANDLE hHandle Handle received by 'Open' function for target printer DWORD dwTxLen The length of the data to be sent to the printer LPBYTE lpbTxBuff Data to be sent to the printer LPDWORD lpdwRxLen The expected length of the data to be received from the printer DWORD dwTimeout The length of the data received from the printer.
Output	LPBYTE lpbRxBuff Data received from the printer if lpdwRxLen is not NULL LPDWORD lpdwRxLen The actual length of the data received from the printer
Description	This function sends data to printer as a real time command If a request was made, receives the data from the printer
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_RESPONSE Response from the printer is invalid. DRV_ERR_CONNECTION_TIMEOUT The printer is not connected or power off No response is returned from the printer in a set time DRV_ERR_CONNECTION_NOT_OPENED TPPIOpen() was not called previously DRV_ERR_INVALID_PARAMETER Input parameter is not correct DRV_ERR_PRINTING_ALREADY_IN_PROGRESS Print job is already sent to the printer by GDI DRV_ERR_INTERNAL Error occurred between the API and the language monitor
Comments	• If lpbTxBuff is NULL, function returns "DRV_ERR_INVALID_PARAMETER" • If dwTXLen is NULL, function returns "DRV_ERR_INVALID_PARAMETER" • If lpdwRxLen is not NULL and no reponse is received after exceeding dwTimeout, this function return with lpdwRxLen=0 and a specific error • Printer driver does not check what command or printing data that is sent to the printer through this function. • If the length of expected reply is longer than the actual reply of the command, it will return immediately after the response. Function returns "DRV_SUCCESS" • If the length of expected reply is shorter than the actual reply of the command, it will return only the buffer size. Function returns "DRV_SUCCESS"

8.5.6 TPPIPartialCut_Declare

Function Name	TPPIPartialCut_Declare
Function Prototype	int TPPIPartialCut_Declare (HANDLE hHandle);
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function enables partial cut to be performed when using 'Partial Cut Perform'
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer
Comments	This function is required to be completed successfully before 'Partial cut Perform' is enabled This declaration will end when a FULL CUT is performed.

8.5.7 TPPIPartialCut_Perform

Function Name	TPPIPartialCut_Perform
Function Prototype	int TPPIPartialCut_Perform (HANDLE hHandle);
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function performs a partial cut to the document
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer
Comments	To perform a partial cut, Partial Cut Declare must be called first. If the Partial Cut declaration is not valid, this function will perform a full cut instead.

8.5.8 TPPIPartialCut_FeedCut

Function Name	TPPIPartialCut_FeedCut
Function Prototype	int TPPIPartialCut_ FeedCut (HANDLE hHandle, BYTE bFeedLines);
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bFeedlines **[0-255]** Number of lines to feed before performing partial cut
Output	None
Description	This function will feed *bFeedlines* amount of lines, and then perform a partial cut to the document
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct DRV_ERR_INTERNAL Error occurred between the API and the language monitor
Comments	Partial cut declaration does not need to be called first to enable this partial cut

8.5.9 TPPIRetract_Eject

Function Name	TPPIRetract_Eject
Function Prototype	int TPPIRetract_Eject (HANDLE hHandle);
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function will perform an Eject action when the paper is ready in the presenter
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.10 TPPIRetract_Retract

Function Name	TPPIRetract_Retract
Function Prototype	int TPPIRetract_Retract (HANDLE hHandle);
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function will execute a retract action when the paper is ready at the presenter
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.11 TPPIRetract_RetryCount

Function Name	TPPIRetract_RetryCount
Function Prototype	int TPPIRetract_RetryCount (HANDLE hHandle, BYTE bRecount);
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bRecount **[0-3]** – Configuration value is *bRecount + 2 tries* The number of retries it will do to retract/eject that will be performed automatically
Output	None
Description	This function will set the number of retract/eject retries it will perform in cases where retract/eject is called.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.12 TPPIRetract_Timeout

Function Name	TPPIRetract_Timeout
Function Prototype	int TPPIRetract_Timeout (HANDLE hHandle, BYTE bReTimeout);
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bReTimeout **[0-4]** – Configuration value is *bReTimeout x 20secs* The number of seconds before retract/eject is performed
Output	None
Description	This function will set the number of seconds it will wait before performing Retract/Eject.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.13 TPPIBezel_Opt

Function Name	TPPIBezel_Opt
Function Prototype	int TPPIBezel_Opt (HANDLE hHandle, BYTE bBezelOpt);
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bBezelOpt **[1-32]** The option for bezel LED configuration that will be used
Output	None
Description	This function will let the user choose the Bezel LED configuration
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.14 TPPIBezel_AutoMode

Function Name	TPPIBezel_AutoMode
Function Prototype	int TPPIBezel_AutoMode (HANDLE hHandle);
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function change the Bezel LED settings to Auto Mode
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.15 TPPIBezel_Usermode

Function Name	TPPIBezel_UserMode
Function Prototype	int TPPIBezel_AutoMode (HANDLE hHandle);
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function change the Bezel LED settings to User Mode
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	N/A

8.5.16 TPPIBezel_Configure

Function Name	TPPIBezel_Opt
Function Prototype	int TPPIBezel_Opt (HANDLE hHandle, BYTE bK, BYTE bL, BYTE bM, BYTE bN, BYTE bP1, BYTE bP2, BYTE bP3, BYTE bP4, BYTE bQ, BYTE bR, BYTE bS);
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bK **[0 – 2]** 0: Auto Mode 1: User Mode 2: Reset all configuration to default BYTE bL **[1 – 32]** *Auto Mode: Specifies the state number of the printer User Mode: Specifies ID to register the blink pattern of Bezel LED **Refer to Appendix 1 for list of state numbers* BYTE bM **[8bit data]** xxxx xRAG - Specifies the color of Bezel LED R – Red LED A – Amber LED G – Green LED BYTE bN **[1-32]** Specifies the number of bits that will be used for the blink pattern defined by bP1-bP4. BYTE bP1-bP4 **[8bit data for each byte]** 0: LED OFF 1: LED ON **blink pattern starts from bp1 to bp4* *bP1(xxxx xxxx) … bP4(xxxx xxxx)* BYTE bQ **[0-255]** – configuration value is *Q x 100ms* Multiplier for the time LED will be ON in the blink pattern BYTE bR **[0-255]** – configuration value is *R x 100ms* Multiplier for the time LED will be OFF in the blink pattern BYTE bS **[0-255]** – configuration value is *S x 100ms* Multiplier for the pause time between blink patterns (LED will be OFF)
Output	None
Description	This function will let the user choose the Bezel LED configuration
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	If any of the parameter is invalid, the command will not be processed

8.5.17 TPPIBuzzerSoundCtrl

Function Name	TPPIBuzzerSoundCtrl
Function Prototype	int TPPIBuzzerSoundCtrl (HANDLE hHandle, BYTE bBuzzerSound)
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bBuzzerSound **[0-2]** Buzzer sound selection ID. Configuration value: 0: Disable 1: Option Buzzer mode 2: Internal Buzzer mode
Output	None
Description	This function will select buzzer sound mode in printer.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	If any of the parameter is invalid, the command will not be processed

8.5.18 TPPIStartBuzzer

Function Name	TPPIStartBuzzer
Function Prototype	int TPPIStartBuzzer (HANDLE hHandle, BYTE bSoundPattern, BYTE bSoundTime)
Input	HANDLE hHandle Handle received by 'Open' function for target printer BYTE bSoundPattern **[1-7]** Select pattern buzzer sound pattern ID. Configuration value are: 1: Pattern A 2: Pattern B 3: Pattern C 4: Pattern D 5: Pattern E 6: Pattern for error 7: Pattern for paper-end BYTE bSoundTime **[0-255]** Specify number of repetitions for the specified sound pattern. Configuration value are: 0: Repeat infinitely 1-255: Repeat based on the setting number.
Output	None
Description	This function is to start sound the buzzer based on pattern and repeat time.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	If any of the parameter is invalid, the command will not be processed

8.5.19 TPPIStopBuzzer

Function Name	TPPIStopBuzzer
Function Prototype	int TPPIStopBuzzer (HANDLE hHandle)
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	None
Description	This function is to stop buzzer sound while buzzer sounding.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_CANNOT_SEND_DATA Printer driver cannot send data to printer
Comments	N/A

8.5.20 TPPIGetInternalBuzzer

Function Name	TPPIGetInternalBuzzer
Function Prototype	int TPPIGetInternalBuzzer (HANDLE hHandle, BYTE bSoundPattern, PINT_BUZZER pIntBuzzer)

Input	HANDLE hHandle
	Handle received by 'Open' function for target printer
	BYTE bSoundPattern **[1-5]**
	Select specify internal buzzer pattern ID. Configuration value are:
	1: Pattern A
	2: Pattern B
	3: Pattern C
	4: Pattern D
	5: Pattern E
	PINT_BUZZER pIntBuzzer
	Structure holds the internal buzzer's pattern configuration. The structure are:
	
	`typedef struct {`
	`  BYTE Header; [always 0x37]`
	`  BYTE Identifier; [always 0x39]`
	`  DWORD Pattern;`
	`  struct {`
	`  DWORD Sound_1;`
	`  DWORD Sound_2;`
	`  DWORD Sound_3;`
	`  DWORD Sound_4;`
	`  DWORD Sound_5;`
	`  DWORD Sound_6;`
	`  }SoundSetting;`
	`  struct {`
	`  DWORD Duration_1;`
	`  DWORD Duration_2;`
	`  DWORD Duration_3;`
	`  DWORD Duration_4;`
	`  DWORD Duration_5;`
	`  DWORD Duration_6;`
	`  }DurationSetting;`
	`}INT_BUZZER,*PINT_BUZZER;`

Output	Output will represent the pattern selection configuration
Description	This function will get the internal buzzer configuration for each pattern.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	If any of the parameter is invalid, the command will not be processed

8.5.21 TPPISetInternalBuzzer

Function Name	TPPISetInternalBuzzer
Function Prototype	int TPPISetInternalBuzzer (HANDLE hHandle, PINT_BUZZER pIntBuzzer)

Input	HANDLE hHandle
	Handle received by 'Open' function for target printer
	PINT_BUZZER pIntBuzzer
	Structure holds the internal buzzer's pattern configuration. The structure are:
	
	`typedef struct {`
	`BYTE Header; [always 0x37]`
	`BYTE Identifier; [always 0x39]`
	`DWORD Pattern; [1-5]`
	`struct {`
	`DWORD Sound_1;`
	`DWORD Sound_2;`
	`DWORD Sound_3;`
	`DWORD Sound_4;`
	`DWORD Sound_5;`
	`DWORD Sound_6;`
	`}SoundSetting; [0-1] 0: OFF, 1: ON`
	`struct {`
	`DWORD Duration_1;`
	`DWORD Duration_2;`
	`DWORD Duration_3;`
	`DWORD Duration_4;`
	`DWORD Duration_5;`
	`DWORD Duration_6;`
	`}DurationSetting; [0-100] 1 unit mean 100ms`
	`}INT_BUZZER,*PINT_BUZZER;`

Output	None
Description	This function will set configuration of internal buzzer based on pattern selected.
Function returns	DRV_SUCCESS Function completed successfully DRV_ERR_INVALID_PARAMETER Input parameter is not correct
Comments	If any of the parameter is invalid, the command will not be processed

8.5.22 TPPIGetInterfaceType

Function Name	TPPIGetInterfaceType
Function Prototype	int TPPIGetInterfaceType (HANDLE hHandle)
Input	HANDLE hHandle Handle received by 'Open' function for target printer
Output	Interface type of printer driver
Description	This function will get the interface type of driver based on printer handle
Function returns	DRV_ERR_CONNECTION_NOT_OPENED TPPIOpen() was not called previously DRV_INTERFACE_USB Printer use virtual printer port for USB DRV_INTERFACE_LPT Printer use parallel port connection (Line Print Terminal) DRV_INTERFACE_LAN Printer use ethernet connection (Local Area Network) DRV_INTERFACE_COM Printer use serial connection (RS232) DRV_INTERFACE_UNKNOWN Printer use local port, print to file or unknown port connection
Comments	If any of the parameter is invalid, the command will not be processed

9. API Application Programming Guide

API application is use to call the printer driver APIs to send any command to printer or get the printer status. User need to pre-install the Toshiba windows printer driver **(TRST-P1X, TRST-P2X, TOSHIBATEC KOP-3X, TOSHIBATEC KOP-3S06 or TRST-L1X)** in PC and **TPPIComm.dll** must be present in **C:\Windows\System32** location.

9.1 Loading API

Whenever **TRST-P1X**, **TRST-P2X**, **TOSHIBATEC KOP-3X, TOSHIBATEC KOP-3S06** or **TRST-L1X** printer driver is installed API dll **(TPPIComm.dll)** will be present in "C:\Windows\System32" location. We will load the API dll **"TPPIComm.dll"** from "C:\Windows\System32" location.

This dll basically communicates with printer driver through its API calls and retrieve or send information to/from the printer.

This application will use the **TPPIComm.dll** for calling its APIs. So first step we will load this dll using **LoadLibrary** function where we specify the path of dll. If loading of this dll is success **hGetProcIDDLL** will have the handle for dll.

```
hGetProcIDDLL = LoadLibrary(L"C:\\Windows\\System32\\TPPIComm.dll");
```

9.2 Open API

Below function is use to load TPPIOpen function from TPPIComm.dll

```
TPPIOpen = (Open)GetProcAddress(hGetProcIDDLL, "TPPIOpen");
```

Above step is to get the exported Open API function address by using function **"GetProcAddress".** The first parameter is **hGetProcIDDLL** which is the handle of the **TPPIComm.dll** and second parameter is the function name. If this function is success then we will have the function address of the Open API (**TPPIOpen**).

If we are able to get the address of Open API, Next step is to open the connection between printer and Application by calling **TPPIOpen** API.

```
int iOpen = TPPIOpen((LPWSTR)printerName, &hPrinter);
```

"**printerName**" is the printer model name and "**hPrinter**" is returned handle of the printer

To know more about the **TPPIOpen** API input parameters and output return types please refer *TPPIOpen* section.

9.3 Close API

Below function is use to load TPPIClose function from TPPIComm.dll

```
TPPIClose = (Close)GetProcAddress(hGetProcIDDLL, "TPPIClose");
```

Above step is to get the exported Close API function address by using function **"GetProcAddress".** The first parameter is **hGetProcIDDLL** which is the handle of the **TPPIComm.dll** and second parameter is the function name. If this function is success then we will have the function address of the Close API (**TPPIClose**).

If we have function address of Close API then we call **TPPIClose** to close the connection between printer and Application.

```
int iClose = TPPIClose((HANDLE*)hPrinter);
```

"hPrinter" is handle of the printer

To know more about the **TPPIClose** API input parameters and output return types please refer the TPPIClose section

9.4 GetStatus

Below function is use to load TPPIGetStatus function from TPPIComm.dll

```
TPPIGetStatus = (GetStatus)GetProcAddress(hGetProcIDDLL, "TPPIGetStatus");
```

Above step is to get the exported GetStatus API function address by using function **"GetProcAddress".** The first parameter is **hGetProcIDDLL** which is the handle of the **TPPIComm.dll** and second parameter is the function name. If this function is success then we will have the function address of the GetStatus API (**TPPIGetStatus**).

If we have function address of GetStatus API then we call **TPPIGetStatus** as below

```
int iGetStatus = TPPIGetStatus(hPrinter, &status);
```

The first parameter of this function is printer handle for which we need the current status and second parameter is the status which we will receive as output.

Below is example to get paper low / paper near end status :

Declare **status** structure as below :

```
STS_INTERNAL status;
memset(&status, 0x00, sizeof(STS_INTERNAL));
```

```
if ((status.AllStatus.Bit.PaperLow1) || (status.AllStatus.Bit.PaperLow2))
{
    GetDlgItem(IDC_STATICSTATUS)->SetWindowTextW(L"Paper Low");
}
GetDlgItem(IDC_EDITOUTPUTMESSAGE)->SetWindowText(L"GetStatus Success!");
```

To know more about the **TPPIGetStatus** API input parameters and output return types please refer TPPIGetStatus section

9.5 Command

TPPITxRx API from **TPPIComm.dll** is used to send any command to printer (send or receive data from printer).

Below function is use to load TPPITxRx function from TPPIComm.dll

```
TPPITxRx = (SendCommand)GetProcAddress(hGetProcIDDLL, "TPPITxRx");
```

Above step is to get the exported TxRx API function address by using function **"GetProcAddress".** The first parameter is **hGetProcIDDLL** which is the handle of the **TPPIComm.dll** and second parameter is the function name. If this function is success then we will have the function address of the TxRx API (**TPPITxRx**).

Then we calls the **TPPITxRx** API and sends the converted hex command **(commandToSend)** along with printer **handle** and **commandLength** values. The API return data will be assigned to **receive** parameter. **Size** is the expected printer response size

```
int  iTxRx = TPPITxRx(hPrinter, (batchCommandLength/2), commandToSend, &size, receive, 0);
```

To know more about the **TPPITxRx** API input parameters and output return types please refer TPPITxRx section

9.6 Control Command (USB Only)

Similar to command feature, Control Command will be used to send command and receive Real time status of printer. This feature will call **TPPICtrlTxRx** API from **TPPIComm.dll** and will be used to send Real Time transfer commands.

Below function is use to load TPPICtrlTxRx function from TPPIComm.dll

```
TPPICtrlTxRx = (SendRealTimeCommand)GetProcAddress(hGetProcIDDLL, "TPPICtrlTxRx");
```

Above step is to get the exported CtrlTxRx API function address by using function **"GetProcAddress".** The first parameter is **hGetProcIDDLL** which is the handle of the **TPPIComm.dll** and second parameter is the function name. If this function is success then we will have the function address of the CtrlTxRx API (**TPPITxRx**).

Then we calls the **TPPICtrlTxRx** API and sends the converted hex command **(commandToSend)** along with printer **handle** and **realTimecommandLength** values. The API return data will be assigned to **receive** parameter. **Size** is the expected printer response size

```
int  iCtrlTxRx = TPPICtrlTxRx(hPrinter, (realTimeCommandLength/2), commandToSend, &size, receive, 0);
```

To know more about the **TPPICtrlTxRx** API input parameters and output return types please refer TPPICtrlTxRx section

10. Error Codes

The following list describes error codes that are returned by the TPPIxxxx() API's.

Value	Parameter Name	Description
0x00	DRV_SUCCESS	Success
0x03	DRV_ERR_CONNECTION_ALREADY_OPENED	Connection is already opened
0x05	DRV_ERR_CANNOT_SEND_DATA	Driver Language Monitor failed to send data to printer
0x06	DRV_ERR_INVALID_RESPONSE	The response of the printer is invalid.
0x07	DRV_ERR_CONNECTION_TIMEOUT	Printer does not respond within the specific time
0x08	DRV_ERR_CONNECTION_NOT_OPENED	Printer Connection not opened
0x09	DRV_ERR_INVALID_PARAMETER	Input Parameter is invalid
0x0B	DRV_ERR_PRINTING_ALREADY_IN_PROGRESS	Print job is already sent to the printer by GDI
0x0C	DRV_ERR_NO_PAPER	No paper found
0x0D	DRV_ERR_FILE_NOT_FOUND	Specified file not found
0x0E	DRV_ERR_INVALID_FILE_FORMAT	The format of the file is invalid or the file has wrong CRC
0x10	DRV_ERR_INTERNAL	Error between the DLL and the language monitor
0x19	DRV_ERR_SPOOL_SUSPEND	Spool service suspended
0x28	DRV_ERR_PRINTER_CANNOT_OPEN	Printer handle can not open

11. Restrictions and Cautions

11.1 Not to unplug connection before printing job done

If user unplug connection before printing job done, the printing job might be stuck in job que list. This can stop next printing job execution. User need to delete it manually

11.2 Page setup setting in the printing document application is affect print result

- Font Substitution Edit ControlA

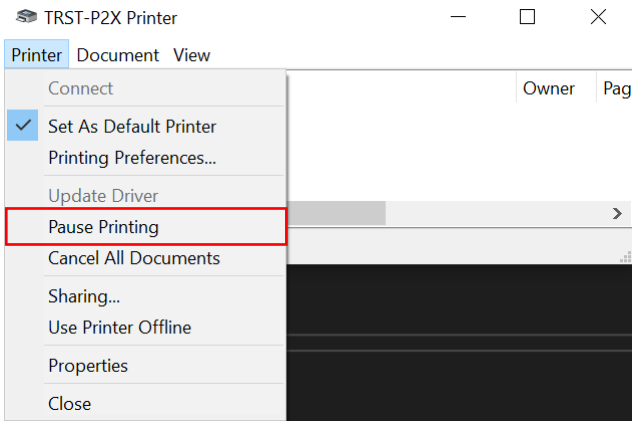
Printer is still print blank page when top margin is set up and print page numbers setting is enabled. Also need to make sure the control characters in the printing application is located in the printable area of the printer (default margin for left and right side is 4mm)

- **Document Printing**

Make sure the page margin setting in the print application is cover the image area to prevent any unexpected printing result (default margin setting for left and right side is 4mm)

11.3 Pause print job in print queue

Pause print job mean spooler to stop sending data to printer. Pause only can perform before sending the print job. Can find the function in ***Print Queue > Printer > Pause Printing***.



11.4 Auto paper size check box

- **Paper guides illegal set**

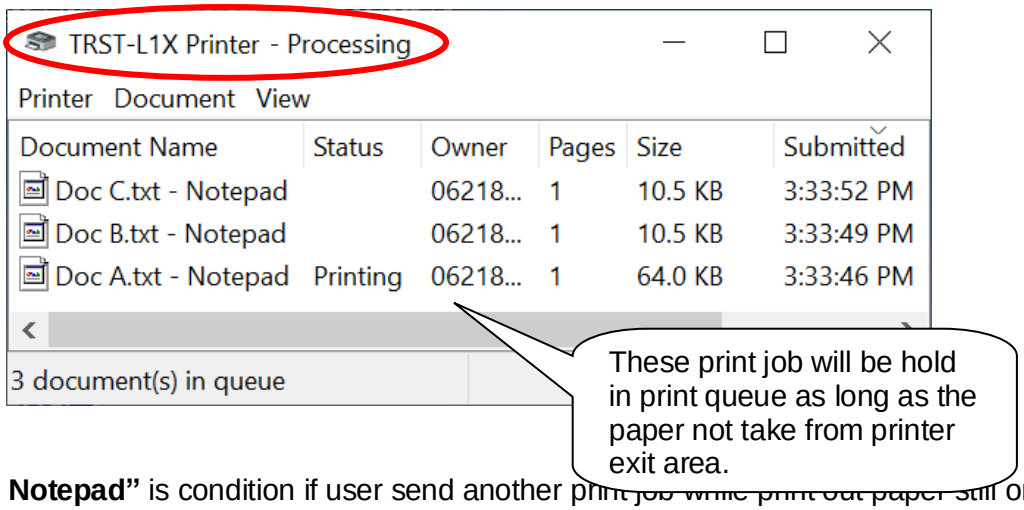
If the paper guides are set to an invalid position, the paper size will default to the system's default setting (80 x 3200 mm). For reference on illegal paper guide positions, ***see page 57 (C position).***

- **Re-open printer UI**

After adjusting the paper guides, users must close and re-open the Printing Preferences and Printer Properties UI to ensure the paper size settings are updated correctly.

11.5 Paper removal function (Only for TRST-L1X printer)

Paper removal function will enable after printer cut is executed and have paper on the printer exit area. If user send another print job in within this condition, the new print job will be hold in the print queue and print queue header change to "**Processing**":



"Doc A.txt – Notepad" is condition if user send another print job while print out paper still on the printer exit area, **"Doc B.txt – Notepad"** and "**Doc C.txt – Notepad"** are conditions if user keep sending print job. The print job will be print immediately after paper is take out from printer exit area.

Note:

For paper removal function RS232 and Ethernet can't support multiple pages in one print queue.

11.6 Changing printer model from KOP-3S01 to KOP-3S01-A or vice versa

After changing the printer model, users must restart the PC or operating system. This step ensures that the default paper size is updated automatically based on the selected printer model.

Furthermore, when performing a test print from the printer properties, the settings may initially reflect the previous paper size configuration. To refresh and apply the updated settings, users should print the test page twice.