



# VB.NET Software Development Kit

How to use StarIO in VB for .NET

Thermal Line Mode Printing

This SDK contains a VB.NET Visual Studio 2005 project for use on Windows XP, Vista, 7, 8, & 8.1. Modern UI and Windows RT are not supported.

|  |   |
|--|---|
| <p>Works with these Printer Model Series:</p> <ul style="list-style-type: none"> <li>FVP10 (Ver.1.0 or later)</li> <li>HSP7000 (Ver.1.0 or later)</li> <li>TSP650 (Ver.2.0 or later)</li> <li>TSP650II (Ver.1.0 or later)</li> <li>TSP700II (Ver.2.0 or later)</li> <li>TSP800II (Ver.1.0 or later)</li> <li>TUP500 (Ver.1.0 or later)</li> <li>TUP900 (Ver.1.2 or later)</li> <li>SP700 (Ver.1.0 or later)</li> </ul> <p>Works with these DK-AirCash Model Series:</p> <ul style="list-style-type: none"> <li>SAC10 (Ver.1.0 or later)</li> </ul> | <p>Supported Interfaces:</p> <ul style="list-style-type: none"> <li>Serial</li> <li>Parallel</li> <li>USB</li> <li>Ethernet</li> <li>Bluetooth</li> </ul> <p>Functions include:</p> <ul style="list-style-type: none"> <li>Print Sample Receipt (EN and JP)</li> <li>1D Barcodes: Code39,93,128, &amp; I2of5</li> <li>2D Barcodes: QR Code &amp; PDF417</li> <li>Drawer Kick</li> <li>Check Block (ETB)</li> <li>Text Formatting</li> <li>Getting Status</li> <li>DK-AirCash</li> </ul> |
|--|---|

Requirements: Visual 2005 or later and .NET Framework 2.0 or later.

## NOTE:

- This sample project contains StarIO components from StarIO Version 2.0.0.0. Details of StarIO(Restrictions, Precautions) are found in the manuals located here:  
 English : <..\StarIO\_help\en\StarIO\index.htm>  
 Japanese : <..\StarIO\_help\ja\StarIO\index.htm>
- This sample project provides source code which tells how to use StarIO components. Executable for 32/64-bit can be built by this project.
- Executable files for 64-bit cannot be executed in a 32-bit environment.
- When you open this project in Windows Vista, 7, 8 or 8.1, execute Visual Studio as an administrator before opening the sln file. You can achieve this by right-clicking on the Visual Studio 2005 icon and clicking "Run as administrator" in the menu displayed.

## Table of Contents




- ❖ [About this Manual](#)
- ❖ [Star Printer Compatibility Chart](#)
- ❖ [How to compile and run the VB.NET SDK](#)
- ❖ [Using the SDK with a Star Micronics Printer](#)
  - [Port Name and Interface Relation](#)
- ❖ [Overview of how the VB.NET SDK is designed](#)
- ❖ [StarIO - \(StarIOPort.DLL\)](#)
  - [How to include StarIO into your project](#)
  - [Configuring your project for 32- or 64-bit](#)
  - [StarIO Methods Overview](#)
    - [Class Variables](#)
      - [PortName](#)
      - [portSettings](#)
      - [Timeout](#)
    - [GetPort](#) - Opening the port to the printer
    - [ReadPort](#)
    - [ReleasePort](#) - Closing the port to the printer
    - [WritePort](#) - Writing data (print job) to the printer
    - [ETB](#) (Verifying the data transmission)
      - [BeginCheckedBlock](#)
      - [EndCheckedBlock](#)
    - [ResetDevice](#)
    - [GetOnlineStatus](#)
    - [GetParsedStatus](#)
    - [Status Class Structure](#)
- ❖ [Functionality](#)
  - [StarIO Printer Commands](#)
    - [Sample Receipts](#)
      - [Print Sample Receipt](#)
      - [Print Sample Receipt \(+ Error Recovery\)](#)
    - [1D Barcodes](#)
      - Code 39
      - Interleaved 2 of 5
      - Code 93
      - Code 128
    - [2D Barcode](#)
      - [QR Code](#)
      - [PDF417](#)
    - [Change Font](#)
    - [Feed](#)

- [Cut](#)
  - [Partial Cut](#)
  - [Full Cut](#)
- [Text Formatting](#)
  - [Slashed Zero](#)
  - [Underline](#)
  - [Upperline](#)
  - [Emphasized \(Bold\)](#)
  - [Upside-Down](#)
  - [Invert Color \(B/W\)](#)
  - [Character expansion](#)
    - [Width](#)
    - [Height](#)
  - [Set Left Margin](#)
  - [Set Right Margin](#)
  - [Alignment](#)
    - [Left](#)
    - [Center](#)
    - [Right](#)
- [SBCS Code Pages](#)
- [Stored Logo Printing](#)
- [Getting Online Status of the Printer](#)
- [Getting Parsed Status of the Printer](#)
- [DK-AirCash](#)
- ❖ [Tips for software application development when using StarIO](#)
  - [Classes](#)
  - [Key Terms](#)
  - [Hexadecimal Dumping Mode](#)
  - [The StarIO Convenience](#)
  - [Additional Features](#)
  - [Communication Options](#)
- ❖ [Additional Resources](#)
  - Star Micronics Developers Network
    - Star Micronics Developers Network
    - Finding Star Micronics Drivers and Additional Sample Code
    - Useful Websites
    - Technical Questions/Support
  - [ASCII Table](#)

## About this Manual

This manual is designed to help you understand StarIO and how to build a VB.NET application to interact with Star Micronics Thermal Line Mode Printers. It is important to understand the basics of the VB.NET language and the .NET framework. Although this SDK is for the programming language VB.NET, there are other SDKs available at our website in the Developers section. Check the Developers section of our site for the newest SDKs, technical documentation, FAQs, and much more additional resources.

### Key Legend:

|                         |   |   |
|-------------------------|---|---|
| <i>Warning</i>          |    | Explains potential issues               |
| <i>Avoid Doing This</i> |   | Explains things not to do               |
| <i>Note</i>             |  | Provides important information and tips |

### CAUTION:

- The information in this manual is subject to change without notice.
- STAR MICRONICS CO., LTD. has taken every measure to provide accurate information, but assumes no liability for errors or omissions.
- STAR MICRONICS CO., LTD. is not liable for any damages resulting from the use of information contained in this manual.
- Reproduction in whole or in part is prohibited.

## Star Printer Compatibility Chart

| Star Printer | SampleReceipt | SampleReceipt (Error Recovery) | Check Block - ETB | Monitoring Status | Open Cash Drawer | 1D Barcodes | 2D Barcodes | Code Page | Font | Feed | Text Formatting | DK-AirCash |
|--------------|---------------|--------------------------------|-------------------|-------------------|------------------|-------------|-------------|-----------|------|------|-----------------|------------|
| FVP10        | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓           | ✓         | ✓    | ✓    | ✓               |            |
| TSP650       | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           |             | ✓         | ✓    | ✓    | ✓               |            |
| TSP650II     | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓           | ✓         | ✓    | ✓    | ✓               |            |
| TSP700II     | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓           | ✓         | ✓    | ✓    | ✓               |            |
| TSP800II     | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓           | ✓         | ✓    | ✓    | ✓               |            |
| TUP500       | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓           | ✓         | ✓    | ✓    | ✓               |            |
| TUP900       | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓<br>*1     | ✓         |      | ✓    | ✓               |            |
| SP700        | ✓             | ✓                              | ✓                 | ✓                 | ✓                |             |             | ✓         | *2   | ✓    | ✓               |            |
| HSP7000      | ✓             | ✓                              | ✓                 | ✓                 | ✓                | ✓           | ✓           | ✓         | ✓    | ✓    | ✓               |            |
| SAC10        |               |                                |                   | ✓                 | ✓                |             |             |           |      |      |                 | ✓          |

\*1 QR Code is not supported. PDF417 is available from F/W ver.3.1 or later.

\*2 Sample code of this SDK is for Thermal Printer command.

When use SP700, please refer "Star Impact Printer Command Specifications".

## How to compile and run the VB.NET SDK

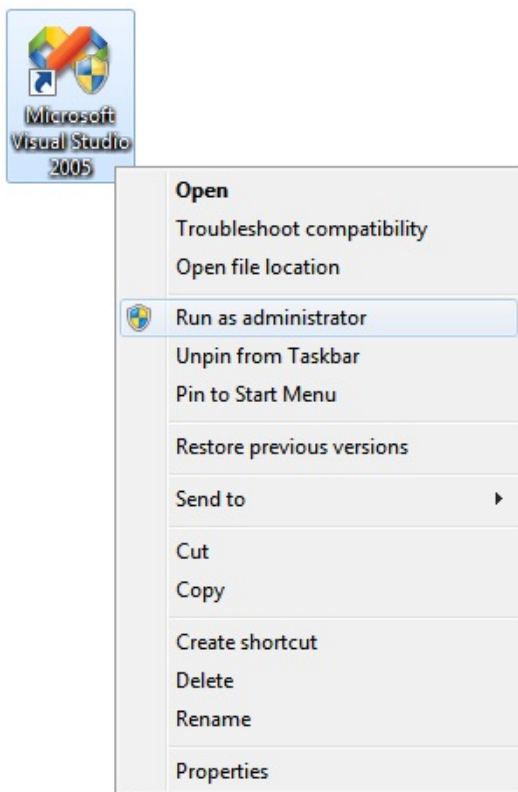
This section will explain:

1. How to open the Visual Studio 2005 VB.NET SDK project.
2. Compiling the project.
3. Running the project.

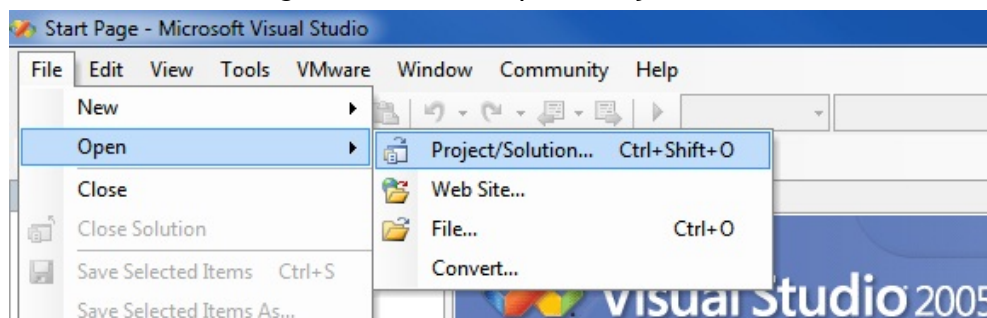
How to open the Visual Studio 2005 VB.NET SDK project:

In Windows XP, open Visual Studio 2005.

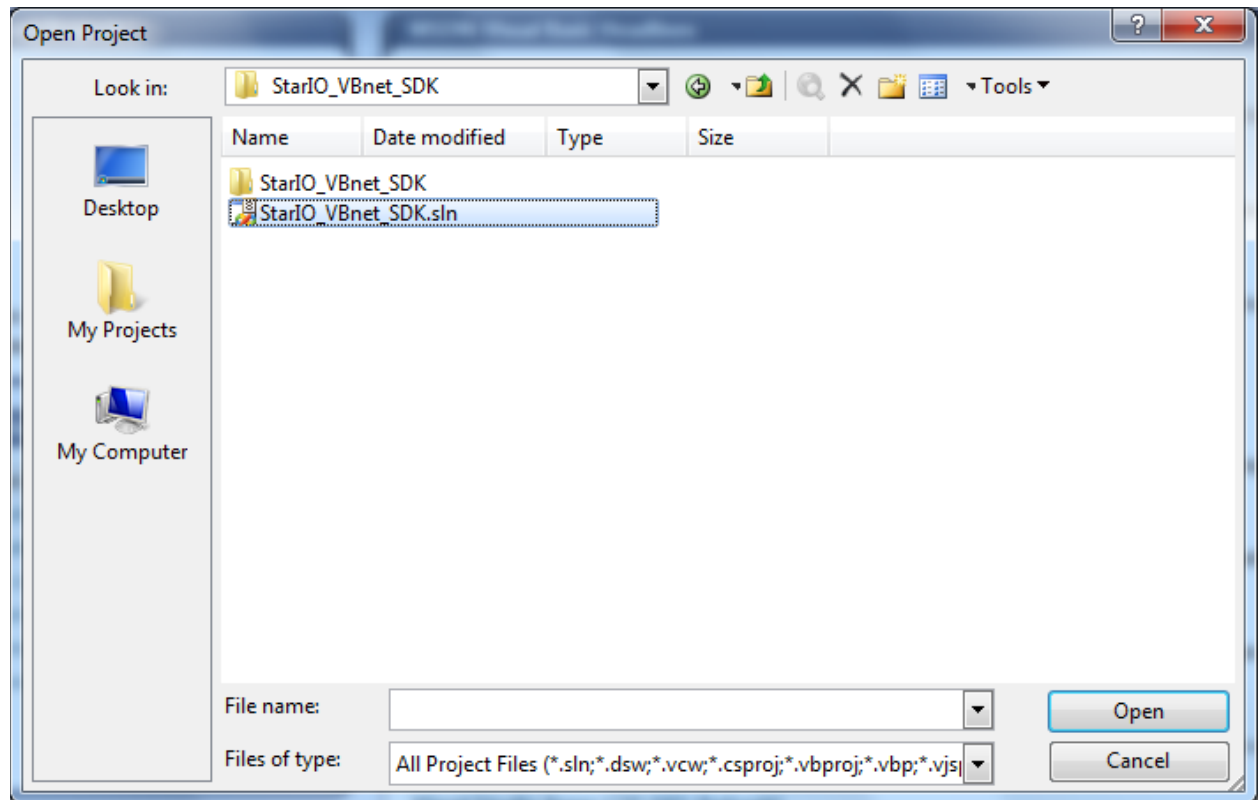
In Vista, 7, 8 or 8.1, right click on Visual Studio 2005 icon and click “Run as administrator”.



Once Visual Studio is running, click on File->Open->Project/Solution...

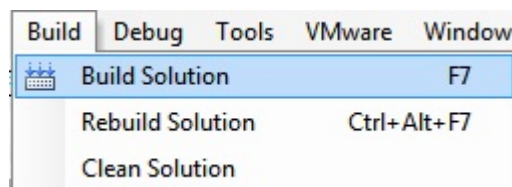


Navigate to the VB.NET SDK folder titled “StarIO\_VBnet\_SDK” and click on the “sln” file titled “StarIO\_VBnet\_SDK.sln” to open the SDK project.



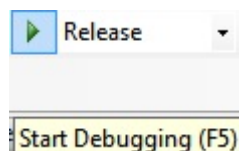
### Compiling the project:

Click on the menu item “Build” and then click “Build Solution” or hit F7



### Running the project:

Click on the green arrow to “Start Debugging” or hit F5



## Using the SDK with Star Micronics Printers

Please make sure you have a compatible Star Micronics Thermal Line Mode Printer model.

### Port Name and Interface Relation:

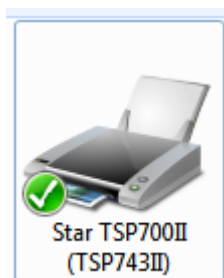
StarIO uses specific port names to identify what port will be used. These are very important to understand because not following the naming convention correctly will fail to communicate with the printer.

| Interface           | Port Name           | Port Settings |
|---------------------|---------------------|---------------|
| Serial              | COMn                | 9600,n,8,1,h  |
| Parallel            | LPTn                | N/A           |
| USB (Vendor Class)  | usbven:             | N/A           |
| USB (Printer Class) | usbprn:"Queue Name" | N/A           |
| Ethernet (TCP/IP)   | tcp:"IP Address"    | N/A           |
| Bluetooth           | BT: COMn            | N/A           |

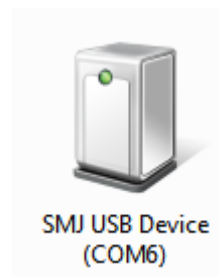


NOTE: If using a **USB** interface and the printer is in **Printer Class Mode**, after successfully installing the printer driver, you will have a Printer Queue Name to use in the Port Name. Use "**usbprn:Star TSP700II (TSP743II)**" as the Port Name.

#### Printer Class Mode



#### Vendor Class Mode



If using a **USB** interface and the printer is in **Vendor Class Mode**, a Port number is not required. Just put "usbven:" as the Port Name.

"LPTn" n = your port number (1, 2, 3, 4 etc)

"COMn" n = your port number (1, 2, 3, 4 etc)

"tcp:192.168.222.244" Enter TCP IP Address of the Ethernet printer.

"BT:COMn" n = your Bluetooth virtual serial port number (1, 2, 3, 4 etc)

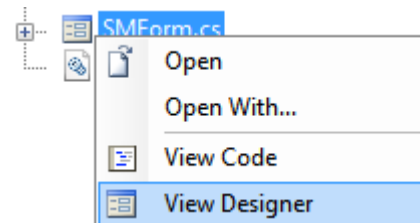


## Overview of how the VB.NET SDK is designed

This overview will touch briefly on key components of the SDK and how to find them.

Focus on the file “SMForm.cs” which contains all the business logic and StarIO commands.

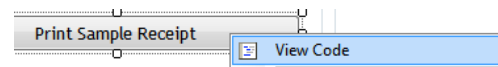
To navigate to a specific command is easy; simply right click on the file called “SMForm.cs” in the solution explorer. Then click on the option “View Designer” which will open the designer.



Click “View Designer”

So let us say for example you wanted to find out what code block is executed for the “Print Sample Receipt” button.

All you need to do is double click on the button in the designer view which will switch the content from designer to code view and take you to the code block handling the mouse click event for the “Print Sample Receipt” code.



Right Click Button, then click “View Code”

With this tip known, you can now fully explore all of the advanced StarIO commands with ease.

Please note that there is a stopping point in the code where everything below the line “CODE BELOW THIS POINT IS NOT USEFUL AS SDK EXAMPLES” is not really useful as code samples because they are related specifically to this SDK program. This code is mostly for the UI so please do not get confused into including this code into your business application.

```
// ~~~~~
// Sample Receipt
// ~~~~~
/// <summary>
/// This function will print a sample receipt to your Star
/// For command codes and examples on how to use spe
/// look at the sample receipt string and reference the
/// You can find these command codes and more in the
/// To access this documentation, open the SDK Readme
/// </summary>
private void cmdPrint_Click(object sender, EventArgs e)
{
    string receipt = "\x1b\x1d\x61\x1" +
        "\x5B" + "If loaded.. Stored Logo 1 goes here" + "\x5D\r" +
        "\x1B\x1C\x70\x1\x0" + //Store
        "Star Micronics Receipt" +
}
```

Visual Studio takes you to the event code block that holds the StarIO commands.



This SDK contains a folder named “Documentation” which contains helper RTF documents for the program to load into the right help window. If these files / folder are moved and/or renamed from this folder, the RTF files will not get loaded correctly and the SDK will not display this helpful info.

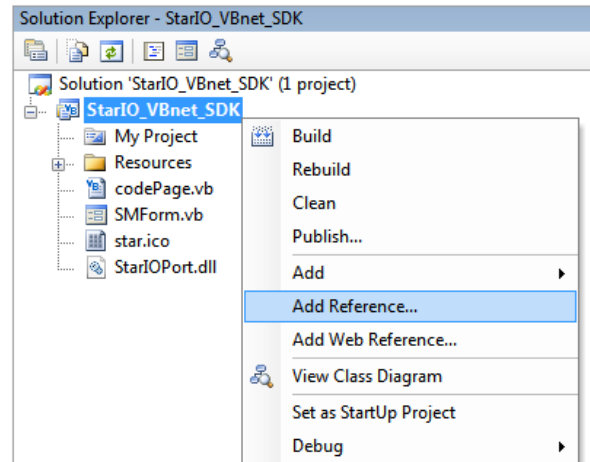
## StarIO - (StarIOPort.dll & StarIO.dll)

### How to include StarIO into your project:

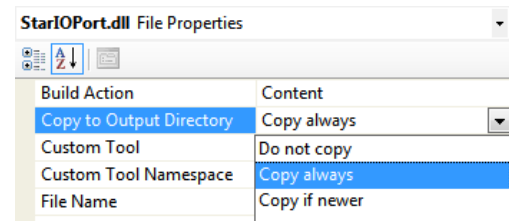
The file StarIOPort.dll is a dynamically linking library that you can include into your VB.NET projects to expose StarIO methods. The file StarIO.dll is a .NET wrapper for StarIOPort.dll.

To include this DLL into your project:

1. Right-Click on the Project Name in the Solution Explorer like this ->
2. Choose Add Reference
3. Click the “Browse” tab
4. Click the “Dependencies” folder attached with this SDK and select the folder “x86” or “x64” based on your target platform.
5. Click “StarIO.dll” to add the first DLL
6. In the Solution Explorer, select StarIO under references. Set the Copy to Output - Property of StarIO.dll to Copy always.
7. StarIOPort.dll should reside in the same folder as StarIO.dll to work. To make sure your project copies the DLLs to your output debug folder, right click on the project name again and select “Add” -> “Existing Item...” then select show all files and select StarIOPort.dll
8. To ensure output debug directory is coping the DLL when you build, click on the item for StarIOPort.dll and under properties, select “Copy to Output Directory” and set to copy always.
9. To expose StarIO, add  
“Imports StarMicronics.StarIO”  
at the top of your main code.
10. Now you can access all of StarIO’s methods!



### How to add StarIO.DLL as a Reference



*Set StarIOPort.dll to always copy*

*Anytime you compile your code with these settings, your executable will also have StarIO.dll & StarIOPort.dll along side of it.*

*Remember that your deliverable executable needs these DLLs along side of it to work.*



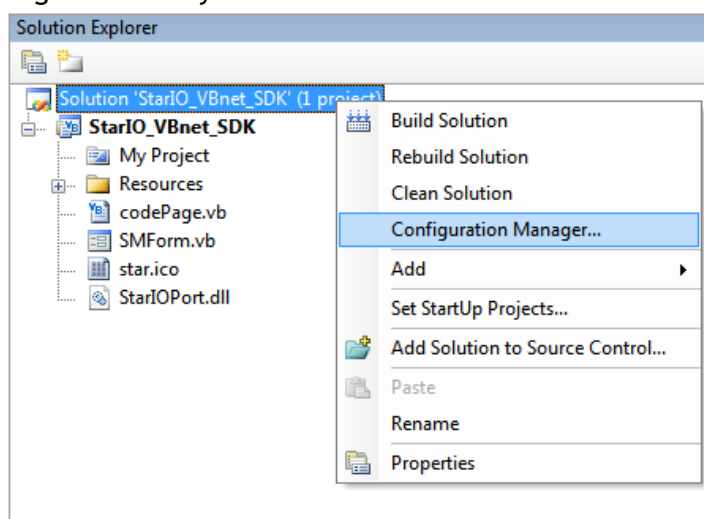
**WARNING:** Make sure StarIO.dll and StarIOPort.dll are in the same directory as each other. StarIO.dll links itself to StarIOPort.dll by looking in the same folder that the StarIO.dll is in.

### Configuring your project to x64 or x86 with StarIO:

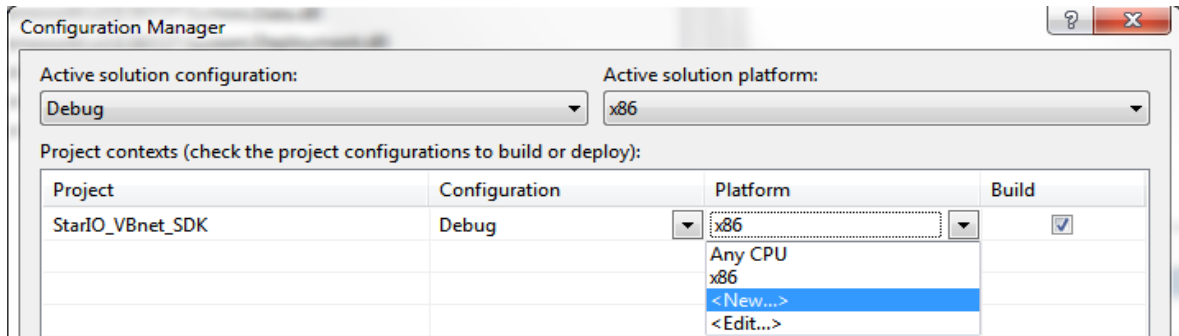
| EXE                   | Runs on x86? | Runs on x64?      | Can use x86 StarIO.dll? | Can use x64 StarIO.dll? | Can use "Any CPU" StarIO.dll? | Can use x86 StarIOPort.dll? | Can use x64 StarIOPort.dll? |
|-----------------------|--------------|-------------------|-------------------------|-------------------------|-------------------------------|-----------------------------|-----------------------------|
| x86 VB .Net Exe       | Yes, in CLR  | Yes, in WOW64+CLR | Yes                     | No                      | Yes                           | Yes                         | No                          |
| x64 VB .Net exe       | No           | Yes, in CLR       | No                      | Yes                     | Yes                           | No                          | Yes                         |
| "Any CPU" VB .Net exe | Yes, in CLR  | Yes, in CLR       | Only on x86             | Only on x64             | Yes                           | Only on x86                 | Only on x64                 |

Compiling your project with the correct StarIOPort.DLL is very important to get the maximum speed from your CPU. Your main two choices are 32-bit and 64-bit Operating systems. WoW64, which means **Windows 32-bit On Windows 64-bit**, allows you to make a 32-bit application with 32-bit binaries to run on either 32-bit or 64-bit. **We recommend setting your project to x86 to run on both 32 and 64 OS.** If you wish to make a 100% 64-bit project then use StarIO dlls under the folder "x64" for 64-bit libraries of StarIO.

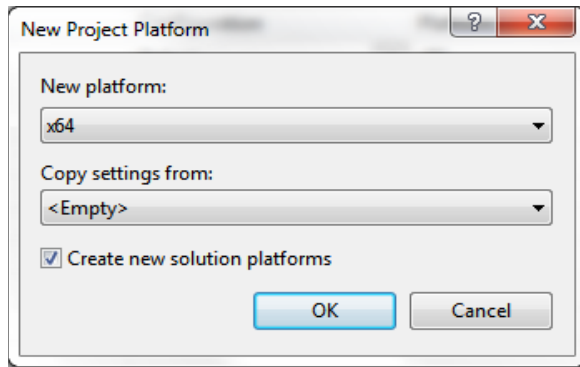
1. Right Click on your VB.NET Solution name and click "Configuration Manager..."



2. Click on the "Platform" drop down box and click "<New...>"



3. Select the platform you wish to compile for (**x86(recommended)** or x64).



4. Click OK and Close. Now that you have selected your platform, you must add 32-bit or 64-bit StarIO DLL.



**WARNING:** If you set your project to “Any CPU” and use 32-bit StarIO libraries, you will find that Windows x64 will not be able to run it. To fix this, setup your project as x86 and use only x86 StarIO which will run on both systems. If you are developing purely for the x64 environment, then set to x64 with x64 StarIO.

5. Once you have selected your target platform and saved your changes, follow the directions in [“How to include StarIO to your project”](#) on how to add a reference to StarIO 32-bit or 64-bit DLL binaries into your project with the corresponding library.

## StarIO Methods Overview:

**Class Variables** include portName (string), portSettings (string), and Timeout (int).



These 3 variables will be “read only” if accessed directly. To assign them use [GetPort\(portName,portSettings,Timeout\);](#) which will allow you to pass in variables to this methods which then assigns the 3 class variables with values.

**portName** is what you will be using to specify the port of communication to the printer.

Ex. “usbven:” “usbprn:TSP650” “tcp:192.168.1.2” “COM4” “LPT1” “BT:COM3”

**portSettings** is what you will use for configuring Serial connections correctly.

Ex. “9600,n,8,1,h”

The following are the acceptable inputs from left to right:

baud: 38400, 19200, 9600, 4800, 2400

parity: n, e, o

data-bits: 8, 7

stop-bits: 1

flow-ctrl: n, h

**Timeout** is a millisecond timeout controlled internally and is used for communication in the APIs (this parameter guarantees that all of the below APIs will complete in a bounded amount of time, but does NOT guarantee the exact timeout length)

### GetPort

```
public static StarIOPort GetPort(String portName, String portSettings, Int TimeoutMillis)
    throws StarIOPortException
```

GetPort is what you will be using to “open” the port to the device. Using one of the valid inputs for portName and portSettings as mentioned previously before this, you can pass your connection string into the StarIO class so that it will correctly set its private variables.

The following would be an actual usage of GetPort in VB.NET:

```
Dim sPort As IPort = Nothing
```

```
Try
```

```
Me.sPort = StarMicronics.StarIO.Factory.I.GetPort(txtPortName.Text, txtPortSetting.Text, 10000)
```

```
'If the port information is bad, catch the error.
```

```
Catch px As PortException
```

```
Me.lblPrinterStatus.Text = "Port Error"
```

```
Return
```

```
End Try
```

**IPort** is a part of StarIO and this will allow you to create a “port” handle. The above example shows the port being created and set to null then being assigned the actual port hook on the following line that contains GetPort.



Always use a **Try**, **Catch** when using **GetPort**. If the port cannot be opened because of connection problems, your program will crash unless you use a **Try**, **Catch** like the above example.

## ReadPort

```
public int ReadPort(Byte[] readBuffer, Int offset, Int size)
```

throws [StarIOPortException](#)

This method reads data from the device. Only use this if you really need to read raw bytes from the device.



**Do not use this method to try and read raw status.**

Use [GetOnlineStatus](#) or [GetParsedStatus](#) for getting status.

### Parameters:

`readbuffer` – A Byte Array buffer into which data is read.

`offset` - specifies where to begin writing data into the `readBuffer[]`

`size` – Total number of bytes to read.

### Returns:

The number of bytes that were actually read. Under some interface types, this function will succeed even when no data was read in. Your application should call this function a limited number of times until the expected data has been read in or until an application determined retry threshold has been reached.

### Throws:

[StarIOPortException](#) - when a communication failure occurs

## ReleasePort

```
public static void ReleasePort(StarIOPort port)
```

This function closes a connection to the port specified.

### Parameters:

`port` - [StarIOPort](#) type representing a previously initialized port.



Always release (close) ports that you get (open).

Leaving a port open will cause future calls to open the port to fail.

## WritePort

```
public int WritePort(Byte[] writeBuffer, Int offset, Int size)
```

throws StarIOPortException

This method writes data to the device. Use this to print to the printer, send commands to the DK-AirCash, etc. The following is an example of how to use this method:

Please keep in mind this is our sample function to send data to the device. The VB.NET SDK has code in printToPrinter that is more complex than this but that code block will show you how to verify data transmission to the device whereas this code is just dumping it:

```
Public Function WritePortHelper(ByVal activePort As IPort, ByVal writeBuffer() As Byte) As Integer
    Dim zeroProgressOccurnaces As Integer = 0
    Dim totalSizeCommunicated As Integer = 0
    While ((totalSizeCommunicated < writeBuffer.Length) And (zeroProgressOccurnaces < 2)) ' adjust
        zeroProgressOccurnaces as needed
        Dim sizeCommunicated As Integer = activePort.WritePort(writeBuffer, totalSizeCommunicated,
CUInt(writeBuffer.Length) - totalSizeCommunicated)
        If (sizeCommunicated = 0) Then
            zeroProgressOccurnaces += CType(1, Integer)
        Else
            totalSizeCommunicated = totalSizeCommunicated + sizeCommunicated
            zeroProgressOccurnaces = 0
        End If
    End While

    WritePortHelper = totalSizeCommunicated
End Function
```

Remember to use a [Try, Catch](#) for safe programming practices.

### Parameters:

`writeBuffer` - Contains the output data in a byte array.

`offset` - Specifies where to begin pulling data from `writeBuffer`.

`size` - Number of bytes to write.

### Returns:

The number of bytes that were actually written. Under some interface types, this function will succeed even when no data was written out. Your application should call this function a limited number of times until all the data has been written out or until an application determined retry threshold has been reached.

**Throws:**

`StarIOException` - when a communication failure occurs

**ETB****BeginCheckedBlock**

```
public StarPrinterStatus BeginCheckedBlock()
```

`throws StarIOException`

This method initiates a checked block printing operation and returns the device's detailed status.

**Returns:**

`StarPrinterStatus` structure giving the current device status - don't bother printing if the printer is offline

**Throws:**

`StarIOException` - when a communication failure occurs

**EndCheckedBlock**

```
public StarPrinterStatus EndCheckedBlock()
```

`throws StarIOException`

This method ends a checked block printing operation and returns the device's detailed status. This function does not return until either the printer has successfully printed all data or has gone offline in error. If the `StarPrinterStatus` structure indicates that the printer is online upon return then all data was successfully printed.

**Returns:**

`StarPrinterStatus` structure giving the current device status - if it's offline then printing failed

**Throws:**

`StarIOException` - when a communication failure occurs



Here is an **example** usage of **BeginCheckedBlock** and **EndCheckedBlock** methods:

```
Try
    Me.sPort = StarMicronics.StarIO.Factory.I.GetPort(txtPortName.Text, txtPortSetting.Text, 10000)
    'If the port information is bad, catch the error.
Catch px As PortException
    'There was an error in opening the port
    Return
End Try

Try
    Me.sPrinterStatus = Me.sPort.BeginCheckedBlock()
Catch ex As PortException
    'Fail: Cannot begin checked block!
    StarMicronics.StarIO.Factory.I.ReleasePort(Me.sPort)
    Return
End Try

If Me.sPrinterStatus.Offline = True Then
    'Fail: The device is offline!
    Factory.I.ReleasePort(Me.sPort)
    Return
End If

Dim writeBuffer As Byte() = ASCIIEncoding.ASCII.GetBytes("ABCD" & vbCrLf)

Dim zeroProgressOccurnaces As UInteger = 0
Dim totalSizeCommunicated As UInteger = 0
Try
    While ((totalSizeCommunicated < writeBuffer.Length) And (zeroProgressOccurnaces < 2))
        Dim sizeCommunicated As UInteger = port.WritePort(writeBuffer, totalSizeCommunicated,
CUInt(writeBuffer.Length) - totalSizeCommunicated)
        If (sizeCommunicated = 0) Then
            zeroProgressOccurnaces += CType(1, UInteger)
        Else
            totalSizeCommunicated = totalSizeCommunicated + sizeCommunicated
            zeroProgressOccurnaces = 0
        End If
    End While
Catch ex As PortException
    'There was an error writing to the port
End Try

Try
    Me.sPrinterStatus = Me.sPort.EndCheckedBlock()
Catch ex As PortException
    'End Checked Block Error
    StarMicronics.StarIO.Factory.I.ReleasePort(Me.sPort)
```

```

    Return
End Try

'check if the device is offline
If Me.sPrinterStatus.Offline = True Then
' the device is offline
    StarMicronics.StarIO.Factory.I.ReleasePort(Me.sPort)
    Return
End If

'Release the port
If Me.sPort IsNot Nothing Then
    StarMicronics.StarIO.Factory.I.ReleasePort(Me.sPort)
End If

```

## ResetDevice

```
public void ResetDevice()
```

throws `StarIOException`

This method resets the device at the hardware level.

### Throws:

`StarIOException` - when a communication failure occurs

## GetOnlineStatus

```
public boolean GetOnlineStatus()
```

throws `StarIOException`

This method returns a Boolean value if the device is online or offline.

### Returns:

Boolean value:      `true` = device is online      `false` = device is offline

### Throws:

`StarIOException` - when a communication failure occurs

An example of its usage:

```

Boolean onlineStatus = port.GetOnlineStatus();

If(onlineStatus){
    //Success! The device is online.
}
Else{
    //Fail! The device is offline.
}

```

## GetParsedStatus

```
public StarPrinterStatus GetParsedStatus ()
```

```
throws StarIOException
```

This method retrieves detailed status form the device with StarIO.

### Returns:

[StarPrinterStatus](#) structure giving the current device status

### Throws:

[StarIOException](#) - when a communication failure occurs

This method uses a class structure that is included with StarIO called [StarPrinterStatus](#)

This structure gives the printer's status in both boolean and binary form.

Create the [StarPrinterStatus](#) object in your project by doing the following:

```
'Create the starIO status variable type
Dim sPrinterStatus As StarPrinterStatus = Nothing
Me.sPrinterStatus = Me.sPort.GetParsedStatus()
'Now parse the returned status for what is flagged as true (or false)
'Status = Offline
If Me.sPrinterStatus.Offline = True Then 'offline == true
    lblPrinterStatus.ForeColor = Color.Red
    printerStatus += "Offline" + Environment.NewLine
End If

'Status = Online
If Me.sPrinterStatus.Offline = False Then 'offline == false
    lblPrinterStatus.ForeColor = Color.Green
    printerStatus += "Online" + Environment.NewLine
End If

'Status = Cover Open
If Me.sPrinterStatus.CoverOpen = True Then
    printerStatus += "Cover Open" + Environment.NewLine
End If

'Paper is empty
If Me.sPrinterStatus.ReceiptPaperEmpty = True Then
    printerStatus += "Paper Empty" + Environment.NewLine
End If

'Status = cash drawer closed
If Me.sPrinterStatus.CompulsionSwitch = False Then
    printerStatus += "Drawer Closed" + Environment.NewLine

'Status = cash drawer open
ElseIf Me.sPrinterStatus.CompulsionSwitch = True Then
    printerStatus += "Drawer Open" + Environment.NewLine
End If

'Status = Paper almost empty
If Me.sPrinterStatus.ReceiptPaperNearEmptyInner = True Then
    printerStatus += "Paper Near Empty" + Environment.NewLine
End If
```

There are lots of different statuses that are pulled when you initialize **StarPrinterStatus**.

**Here is a list of statuses that are in the class structure **StarPrinterStatus**:**

CoverOpen returns a **Boolean**.

Offline returns a **Boolean**.

CompulsionSwitch returns a **Boolean**.

OverTemp returns a **Boolean**.

UnrecoverableError returns a **Boolean**.

CutterError returns a **Boolean**.

MechanicalError returns a **Boolean**.

HeadThermistorError returns a **Boolean**.

ReceiveBufferOverflow returns a **Boolean**.

PageModeCommadError returns a **Boolean**.

BlackMarkError returns a **Boolean**.

PresenterPaperJamError returns a **Boolean**.

HeadUpError returns a **Boolean**.

VoltageError returns a **Boolean**.

ReceiptBlackMarkDetection returns a **Boolean**.

ReceiptPaperEmpty returns a **Boolean**.

ReceiptPaperNearEmptyInner returns a **Boolean**.

ReceiptPaperNearEmptyOuter returns a **Boolean**.

PresenterPaperPresent returns a **Boolean**.

PeelerPaperPresent returns a **Boolean**.

StackerFull returns a **Boolean**.

slipTOF returns a **Boolean**.

slipCOF returns a **Boolean**.

slipBOF returns a **Boolean**.

validationPaperPresent returns a **Boolean**.

slipPaperPresent returns a **Boolean**.

ETBAvailable returns a **Boolean**.

ETBCounter returns a **Byte**.

PresenterState returns a **Byte**.

RawStatus returns a **Byte[]** array.

## StarIO Printer Commands

### StarIO Printer Commands

All of these commands can be found in the [Star Thermal Line Mode Command Manual and Star Impact Printer Command Manual](#).

The VB.NET SDK also has page and section references to this document for more information so please download that manual and study it if you need more detail on a specific command.

### Sample Receipts

The screenshot shows a control panel with the following elements:

- Port Info:**
  - Port Name:
  - Port Setting:
- Print Sample Receipt** (highlighted with a red box):
  - Buttons: Thermal, Dot Matrix
- Print Sample Receipt (+Error Recovery)** (highlighted with a red box):
  - Buttons: Thermal, Dot Matrix

### Print Sample Receipt

Prints an example in English or Japanese. Reference the Text Formatting section of this document for instructions on modifying text.

### Print Sample Receipt (+ Error Recovery)

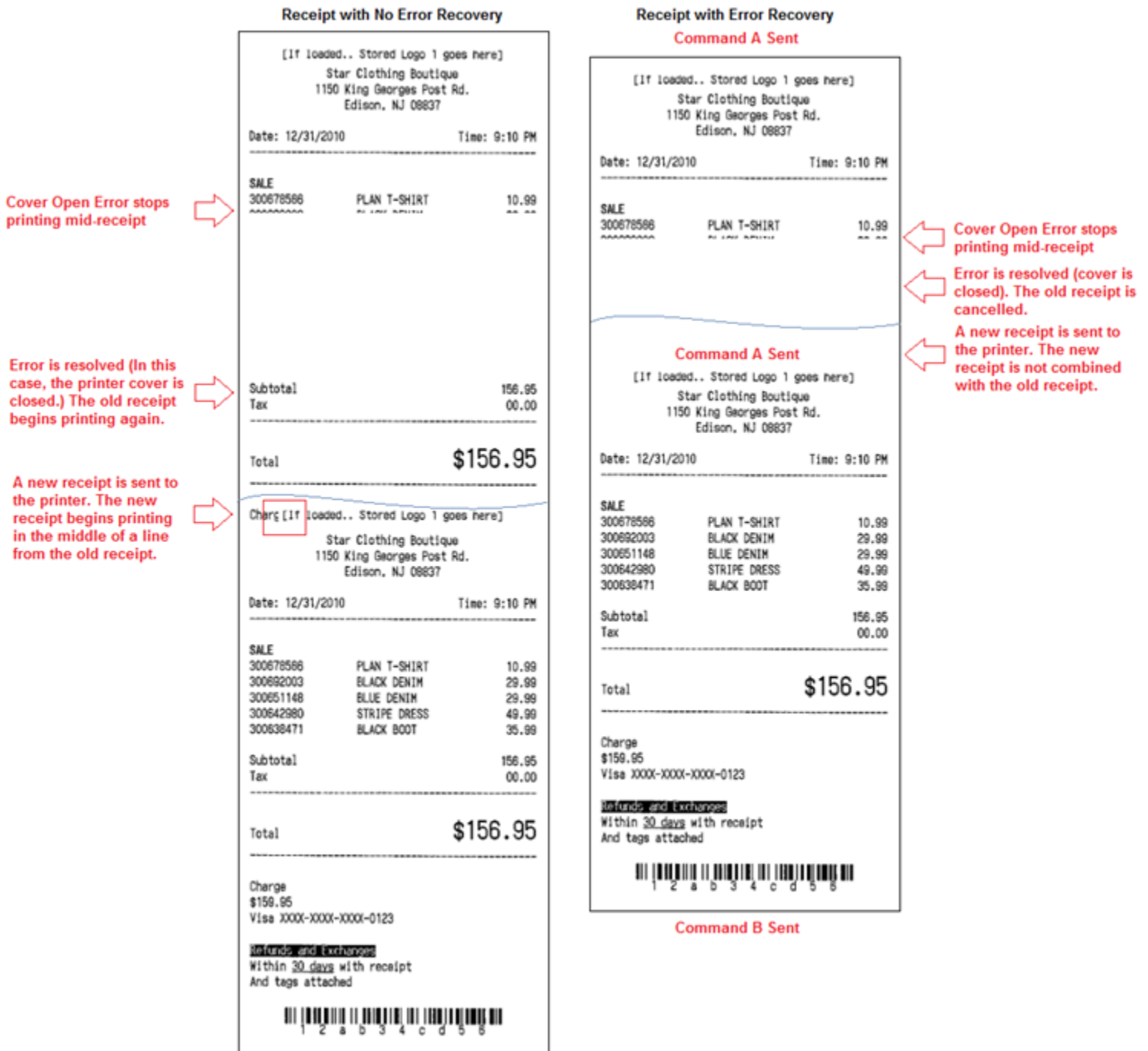
Enables the printer to completely recover from a failed print job due to an error occurring in the middle of printing. For example, the paper runs out in the middle of printing a receipt. The cashier loads another roll of paper and reprints it. Depending on where the old print job was cut short due to paper out, the new receipt may begin in the middle of the old one. Star provides two commands, one to be inserted before the print job and one after, to avoid this issue. A graphic is included on the following page to illustrate this functionality.

Command A: (Add to the head of the receipt data)

ESC GS ETX EOT NUL NUL ESC GS ETX ETX NUL NUL

Command B: (Add to the bottom of the receipt data)

ESC GS ETX EOT NUL NUL



#### Important Notes about This Function:

##### 1. Text Formatting

Command A will release the effect of text formatting commands (ex: bold, underline). Text formatting commands must be added between Command A and Command B for all print data.

##### 2. In the following conditions, the printer will discard all data if Command A is not sent:

- i. Error occurs in sending print data with Command A
- ii. Within ten seconds after an error occurs

(continued from Print Sample Receipt + Error Recovery Notes section)

## 3. Supported printers and minimum firmware version required:

|          |                  |
|----------|------------------|
| FVP10    | Ver.1.2 or later |
| TSP650   | Ver.3.0 or later |
| TSP650II | Ver.1.0 or later |
| TSP700II | Ver.3.0 or later |
| TSP800II | Ver.1.0 or later |
| TUP500   | Ver.3.0 or later |
| SP700    | Ver.3.0 or later |

## 1D Barcodes

ESC b n1 n2 n3 n4 d1 ... dk RS

n1 = Barcode Type

|            |           |              |               |
|------------|-----------|--------------|---------------|
| 0 = UPC-E  | 1 = UPC-A | 2 = JAN/EAN8 | 3 = JAN/EAN13 |
| 4 = Code39 | 5 = ITF   | 6 = Code128  | 7 = Code93    |
|            |           |              | 8 = NW-7      |

n2 = Under-bar character selection and added line feed selection

- 1 = No added under-bar characters & Executes line feed after printing barcode
- 2 = Adds under-bar characters & Executes line feed after printing barcode
- 3 = No added under-bar characters & doesn't line feed after printing barcode
- 4 = Adds under-bar characters & doesn't line feed after printing barcode

n3 = Barcode mode selection specifies the size of the narrow and wide barcode lines


n4 = Barcode height (dot count)

## 2D Barcodes

### QR Codes

QR Code

PDF417



Correction Level

L 7%

Model

Model 2

Cell Size

3

Barcode Data

http://www.StarMicronics.com

Step 1

Step 2

Step 3

Step 4

Step 5

Run All Steps

Set Model

Set Correction Level

Set Cell Size

Set Barcode Data

Print QR Code

Set All and Print

There are 5 commands below that are very important to printing a good QR code.

- |                                  |                                |
|----------------------------------|--------------------------------|
| (1) Set QR Code Model #          | ESC GS y S 0 n                 |
| (2) Set QR Code Correction Level | ESC GS y S 1 n                 |
| (3) Set QR Code Cell Size        | ESC GS y S 2 n                 |
| (4) Set QR Code Data             | ESC GS y D 1 NUL nL nH d1...dk |
| (5) Print the QR Code            | ESC GS y P                     |


Here is the order in which commands need to be sent to the printer for it to print the QR code:

QR Model + QR Correction Level + QR Cell Size + QR Data + Print QR Code



**PDF417**

QR Code PDF417



Barcode Data

Barcode Size

☒ USE\_LIMITS      Height:

☐ USE\_FIXED      Width:

Aspect Ratio      X-Dim ( dots )      Security Level

Please visit page 3-120 in the Line Mode Spec Manual for more details on PDF417

- |  |                                |
|--|--------------------------------|
| (1) Set PDF417 barcode size            | ESC GS x S 0 n p1 p2           |
| (2) Set PDF417 ECC (Security Level)    | ESC GS x S 1 n                 |
| (3) Set PDF417 module X direction size | ESC GS x S 2 n                 |
| (4) Set PDF417 module aspect ratio     | ESC GS x S 3 n                 |
| (5) Set PDF417 barcode data            | ESC GS y D 1 NUL nL nH d1...dk |
| (6) Print PDF417 barcode               | ESC GS x P                     |

Here is the order in which commands need to be sent to the printer for it to print the PDF417.

PDF417 Size + PDF417 ECC + PDF417 X-dim + PDF417 Ratio + PDF417 Data + Print PDF417

## Change Font

Changing the font on the printer can be done with the following commands.

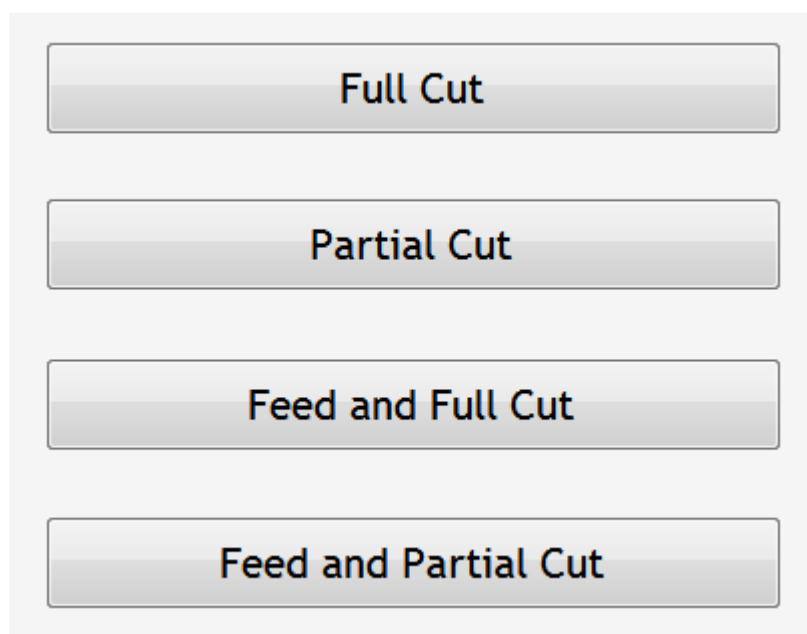
ESC RS F n                      n = 0 for A, 1 for B, 10 for OCR-B

|   |                       |
|---|-----------------------|
| <input type="button" value="Set Font A"/> | Font-A (12 x 24 dots) |
| <input type="button" value="Set Font B"/> | Font-B (9 x 24 dots)  |
| <input type="button" value="Set OCR-B"/>  | OCR-B (16 x 24 dots)  |

## Feed

The feed commands are very straight forward. Use LF for best results.

|  |  |
|--|--|
| <input type="button" value="Line Feed"/>               |  |
| <input type="button" value="Set Line Feed to 4mm"/>    |  |
| <input type="button" value="Set Line Feed to 3mm"/>    |  |
| <input type="button" value="Multi Line Feed"/>         | 4 <input type="button" value="▲"/><br><input type="button" value="▼"/> |
| <input type="button" value="Set Line Spacing to 3mm"/> |  |
| <input type="button" value="Feed 4mm Lines"/>          | 1 <input type="button" value="▲"/><br><input type="button" value="▼"/> |
| <input type="button" value="Feed 8mm Lines"/>          | 1 <input type="button" value="▲"/><br><input type="button" value="▼"/> |
| <input type="button" value="Form Feed"/>               |  |

**Cut**

The interface for the 'Cut' function consists of four vertically stacked buttons. Each button has a light gray background with a subtle gradient and a thin black border. The buttons are labeled 'Full Cut', 'Partial Cut', 'Feed and Full Cut', and 'Feed and Partial Cut' from top to bottom.

Full Cut

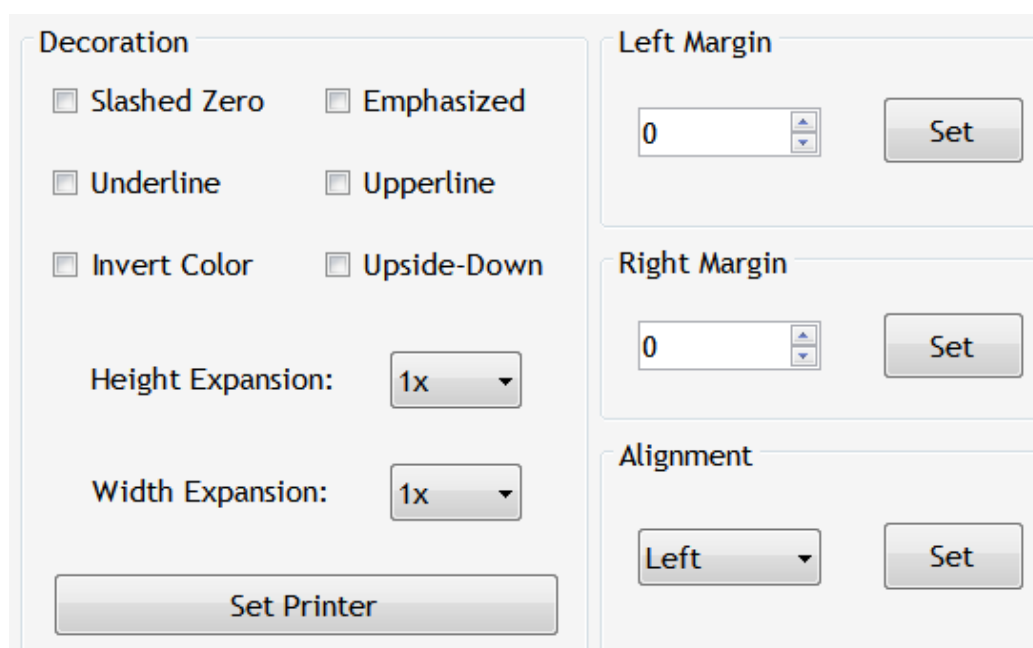
Partial Cut

Feed and Full Cut

Feed and Partial Cut

Partial Cut                      ESC d 1 or 3

Full Cut                        ESC d 0 or 2

**Text Formatting**

The 'Text Formatting' interface is divided into two main sections. The left section, titled 'Decoration', contains six checkboxes: 'Slashed Zero', 'Emphasized', 'Underline', 'Upperline', 'Invert Color', and 'Upside-Down'. Below these are two expansion settings: 'Height Expansion' and 'Width Expansion', each with a dropdown menu currently set to '1x'. A 'Set Printer' button is at the bottom of this section. The right section contains three sub-sections: 'Left Margin' with a numeric input set to '0' and a 'Set' button; 'Right Margin' with a numeric input set to '0' and a 'Set' button; and 'Alignment' with a dropdown menu set to 'Left' and a 'Set' button.

**Decoration**

☐ Slashed Zero      ☐ Emphasized

☐ Underline          ☐ Upperline

☐ Invert Color        ☐ Upside-Down

Height Expansion: 1x ▼

Width Expansion: 1x ▼

Set Printer

**Left Margin**

0      Set

**Right Margin**

0      Set

**Alignment**

Left      Set

The following are all Text Decoration or formatting related.

|                            |                     |                   |
|----------------------------|---------------------|-------------------|
| <b>Slashed Zero</b>        | ESC / n             |                   |
| <b>Underline</b>           | ESC - n             |                   |
| <b>Upperline</b>           | ESC _ n             |                   |
| <b>Invert Color (B/W)</b>  | ESC 4               |                   |
| <b>Emphasized (Bold)</b>   | ESC E = on          | ESC F = off       |
| <b>Upside-Down</b>         | SI = Start          | DC2 = off         |
| <b>Character expansion</b> |                     |                   |
| <b>Width</b>               | ESC W n             | $0 \leq n \leq 5$ |
| <b>Height</b>              | ESC h n             | $0 \leq n \leq 5$ |
| <b>Set Left Margin</b>     |                     |                   |
| ESC l n                    | $0 \leq n \leq 255$ |                   |
| <b>Set Right Margin</b>    |                     |                   |
| ESC Q n                    | $0 \leq n \leq 255$ |                   |
| <b>Alignment</b>           |                     |                   |
| <b>Left</b>                | ESC GS a 0          |                   |
| <b>Center</b>              | ESC GS a 1          |                   |
| <b>Right</b>               | ESC GS a 2          |                   |

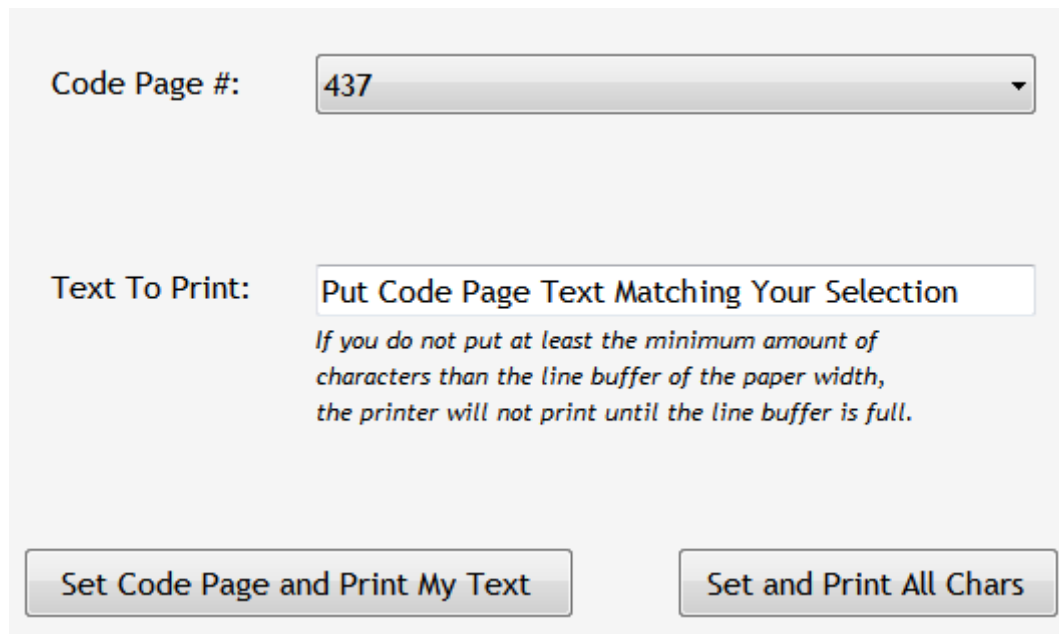
### Code Pages

The code pages supported by Star Printers can be found in the codePage.vb class but please be aware that not all code pages on the printer will be on your PC.

To set a code page on the printer:

ESC GS t n

n = The Code Page Selection Index



The screenshot shows a configuration window with a light gray background. At the top, there is a label 'Code Page #' followed by a dropdown menu showing '437'. Below this, there is a label 'Text To Print:' followed by a text input field containing 'Put Code Page Text Matching Your Selection'. Underneath the input field, there is a note in italics: 'If you do not put at least the minimum amount of characters than the line buffer of the paper width, the printer will not print until the line buffer is full.' At the bottom of the window, there are two buttons: 'Set Code Page and Print My Text' on the left and 'Set and Print All Chars' on the right.

### Stored Logo Printing

Stored logo printing is done in the sample receipt. Please review that and the Thermal Line Mode Manual for more information.

ESC FS p 1 0

### Getting Online Status of the Printer

Visit the function code block called **printToPrinter** and there you will see the StarIO method **GetOnlineStatus** being used to retrieve a boolean value for online status.

True = Online  
False = Offline

### Getting Parsed Status of the Printer

Review the function code block called **timerGetStatus\_Tick** and there you will see the StarIO method **GetParsedStatus** being used to pull a struct down of all the potential status flags the printer can throw. [Click here for the full list of statuses.](#)

## DK-AirCash

Port Info:

|                       |         |                      |     |
|-----------------------|---------|----------------------|-----|
| Printer Port Name:    | usbven: | Drawer Port Name:    | BT: |
| Printer Port Setting: |         | Drawer Port Setting: |     |

Get Printer Status

Get Cash Drawer Status

Sample Receipt + Open Cash Drawer

Open Cash Drawer

### Get Printer Status /Get Cash Drawer Status

The status of the connected printer or cash drawer is displayed.

### Sample Receipt + Open Cash Drawer

When a printer is connected, it prints a sample receipt and a cash drawer is opened if connected. A password is required to open the cash drawer. The default password is "1234"

### Open Cash Drawer

A cash drawer is opened if connected. A password is required to open the cash drawer. The default password is "1234".

DK-AirCash

Please input Password

OK

キャンセル

## Tips for App Development when using StarIO

Star Micronics prides itself as the industry leader in great POS products and with great power comes great responsibility. Below is a tips section just to help you get on the fast track to software development with StarIO.

**TIP #1:** If you are going to be coding a large project, create a class to abstract all the printing methods into class(s) instead of having the code reside in the main code block. This will help with code reusability and will also save you time in the long run from having to find one line of code in the main code. By having StarIO only reside in the class(s), you will be fully taking advantage of object oriented programming.

**TIP #2:** Know what the differences and definitions of (ASCII & Unicode), (Hex & Decimal), and (Byte & Char) A byte is normally 8-bits long which would be 8 digits of binary (1s and 0s). These bytes are just 8 bits of binary data but bytes can also be int or char. The three different variable types pretty much hold the data in the same way but there are slight differences. Try to code with Bytes instead of Chars, ints, or strings when choosing a variable to contain your print job data. ASCII to Unicode and vice versa conversions are sometimes unsecure so make sure you know what and how the encoding class works with these. Big mistakes made in Unicode are culture-sensitive search and casing, surrogate pairs, combining characters, and normalization which are answered [here](#).

**TIP #3:** HEX DUMP MODE! If you are debugging and your application seems to have a bug in it use hex dump mode on the printer. This is the best way to verify what is being sent out of the computer is being received correctly. To put the printer in hex dump mode turn the printer off, open the cover to the paper, hold the feed button down, turn the printer back on, close the cover, let go of the feed button. Hex dump mode is a sure fire way to verify hex data is sent correctly. When in hex dump mode, printer functions will not work.

**TIP #4:** Do not waste time trying to reverse engineer StarIO command codes. All the available StarIO commands are available in the Thermal Line Mode Spec Manual and that is the best resource to use when researching a specific StarIO command. This SDK & Manual was built to help you (The Developer) have a very easy job ahead of you to program for Star Printers.

**TIP #5:** If there is a command that is not covered in this SDK but you wish to see a code snippet of that command in use then visit our developers section for a possible code block that matches your needs.

**TIP #6:** StarIO, ESC/POS, UPOS: JavaPOS, POS for .NET, & OPOS are all different ways to communicate with the printer. Visit our Developers section for more info on these. This SDK covers StarIO only.

## Additional Resources

This section will share resources that will help you develop good software with StarIO.

### [Star Micronics Developers Network](#)

Browse Star Micronics' FAQs, ask a question, look up information, etc.

The Developers Network gets you access to:

- Updated Versions of this Manual and Source Code
- Getting Started Advice and Industry Information
- Star Micronics Printer Drivers
- Technical Questions/Support

### [Download the Star Thermal Line Mode Command Spec Manual](#)

### [Download the Star Impact Printer Command Spec Manual](#)

Use it as your reference for all StarIO Line Mode commands.

### [Character Encoding in the .NET Framework](#)

If you don't know what ASCII and Unicode is, this is a good place to start.

### [Microsoft .NET Internationalization](#)

Good resource for more detail on internationalization.

### [Visual VB.NET Developer Center](#)

Great place to learn more about the VB.NET language.

### [Unicode.org](#)

The Unicode Consortium - Good place to learn more about Unicode.

### [1D Barcodes](#)

Barcode Island is a great resource for specs on 1D barcodes.

### [2D Barcodes](#)

Great place for information on 2D Barcodes, [QR Codes](#), and [PDF417](#)

### [Code Pages](#)

Learn about Code Pages here.



## ASCII Table Resource

| ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol |
|------------------|------------------|------------------|------------------|
| 0 0 NUL          | 16 10 DLE        | 32 20 (space)    | 48 30 0          |
| 1 1 SOH          | 17 11 DC1        | 33 21 !          | 49 31 1          |
| 2 2 STX          | 18 12 DC2        | 34 22 "          | 50 32 2          |
| 3 3 ETX          | 19 13 DC3        | 35 23 #          | 51 33 3          |
| 4 4 EOT          | 20 14 DC4        | 36 24 \$         | 52 34 4          |
| 5 5 ENQ          | 21 15 NAK        | 37 25 %          | 53 35 5          |
| 6 6 ACK          | 22 16 SYN        | 38 26 &          | 54 36 6          |
| 7 7 BEL          | 23 17 ETB        | 39 27 '          | 55 37 7          |
| 8 8 BS           | 24 18 CAN        | 40 28 (          | 56 38 8          |
| 9 9 TAB          | 25 19 EM         | 41 29 )          | 57 39 9          |
| 10 A LF          | 26 1A SUB        | 42 2A *          | 58 3A :          |
| 11 B VT          | 27 1B ESC        | 43 2B +          | 59 3B ;          |
| 12 C FF          | 28 1C FS         | 44 2C ,          | 60 3C <          |
| 13 D CR          | 29 1D GS         | 45 2D -          | 61 3D =          |
| 14 E SO          | 30 1E RS         | 46 2E .          | 62 3E >          |
| 15 F SI          | 31 1F US         | 47 2F /          | 63 3F ?          |

| ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol | ASCII Hex Symbol |
|------------------|------------------|------------------|------------------|
| 64 40 @          | 80 50 P          | 96 60 `          | 112 70 p         |
| 65 41 A          | 81 51 Q          | 97 61 a          | 113 71 q         |
| 66 42 B          | 82 52 R          | 98 62 b          | 114 72 r         |
| 67 43 C          | 83 53 S          | 99 63 c          | 115 73 s         |
| 68 44 D          | 84 54 T          | 100 64 d         | 116 74 t         |
| 69 45 E          | 85 55 U          | 101 65 e         | 117 75 u         |
| 70 46 F          | 86 56 V          | 102 66 f         | 118 76 v         |
| 71 47 G          | 87 57 W          | 103 67 g         | 119 77 w         |
| 72 48 H          | 88 58 X          | 104 68 h         | 120 78 x         |
| 73 49 I          | 89 59 Y          | 105 69 i         | 121 79 y         |
| 74 4A J          | 90 5A Z          | 106 6A j         | 122 7A z         |
| 75 4B K          | 91 5B [          | 107 6B k         | 123 7B {         |
| 76 4C L          | 92 5C \          | 108 6C l         | 124 7C           |
| 77 4D M          | 93 5D ]          | 109 6D m         | 125 7D }         |
| 78 4E N          | 94 5E ^          | 110 6E n         | 126 7E ~         |
| 79 4F O          | 95 5F _          | 111 6F o         | 127 7F □         |

*Use this to compare hex values to symbol (ASCII) values*



Star Micronics is a global leader in the manufacturing of small printers. We apply over 50 years of knowhow and innovation to provide elite printing solutions that are rich in stellar reliability and industry-respected features. Offering a diverse line of Thermal, Hybrid, Mobile, Kiosk and Impact Dot Matrix printers, we are obsessed with exceeding the demands of our valued customers every day.

We have a long history of implementations into Retail, Point of Sale, Hospitality, Restaurants and Kitchens, Kiosks and Digital Signage, Gaming and Lottery, ATMs, Ticketing, Labeling, Salons and Spas, Banking and Credit Unions, Medical, Law Enforcement, Payment Processing, and more!

High Quality POS Receipts, Interactive Coupons with Triggers, Logo Printing for Branding, Advanced Drivers for Windows, Mac and Linux, Complete SDK Packages, Android, iOS, Blackberry Printing Support, OPOS, JavaPOS, POS for .NET, Eco-Friendly Paper and Power Savings with Reporting Utility, ENERGY STAR, MSR Reading, *future*PRNT, StarPRNT... How can Star help you fulfill the needs of your application?

Don't just settle on hardware that won't work as hard as you do. Demand everything from your printer. Demand a Star!

| Version | Release Date |
|---------|--------------|
| 1.0.0   | July 2011    |
| 1.1.0   | Oct. 2011    |
| 1.2.0   | Jan. 2012    |
| 2.0.0   | Mar. 2014    |
| 2.1.0   | July 2014    |

### Star Micronics Worldwide

Star Micronics Co., Ltd.  
536 Nanatsushinya  
Shimizu-ku, Shizuoka 424-0066  
Japan  
+81-54-347-2163  
<http://www.star-m.jp/eng/index.htm>

Star Micronics America, Inc.  
1150 King Georges Post Road  
Edison, NJ 08837  
USA  
1-800-782-7636  
+1-732-623-5500  
<http://www.starmicronics.com>

Star Micronics EMEA  
Star House  
Peregrine Business Park, Gomm Road  
High Wycombe, Buckinghamshire HP13 7DL  
UK  
+44-(0)-1494-471111  
<http://www.star-emea.com>

Star Micronics Southeast Asia Co., Ltd.  
Room 2902C. 29th Fl. United Center Bldg.  
323 Silom Road, Silom Bangrak, Bangkok 10500  
Thailand  
+66-2-631-1161 x 2  
<http://www.starmicronics.co.th/>