



IntelliPay™

V400C Managed DLL

Version 2.1.0.0

Revision Date: June 10, 2021

Prerequisites

- Managed Code Environment
 - Visual Basic
 - .NET
 - C#
- USB Port on Host



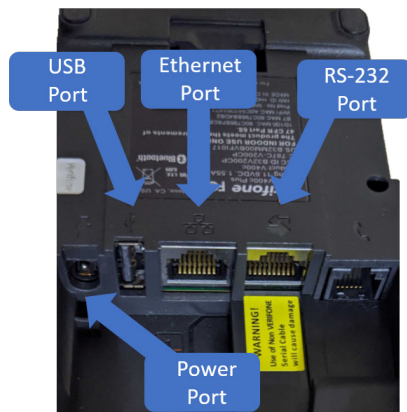
Verifone V400C Terminal

Connection

- Use the USB to RJ50 cable (Provided by IntelliPay) to connect the terminal to the Host
 - RJ50 end plugs into the Ethernet port on the back of the V400C terminal
- Take note of the COM port number that the Host assigns to this terminal



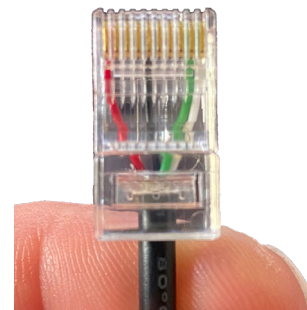
Bottom (back) of V400C



Close-up V400C Connections



IntelliPay supplied USB to RJ50 cable



Close-up of the RJ50 connector end that plugs into the Ethernet port on the Verifone V400C terminal

Implementation

- Import the DLL into your Environment
- You will have three functions in the code
 - SALE
 - VOID
 - REFUND

Open the DLL Connection and Define its event handler

```
C#
IntelliPayDotNet.Terminal terminal = new Terminal("COM6");
terminal.TerminalResponse += new Terminal.TerminalEventHandler(terminal_TerminalResponse);
```

```
VB
Dim terminal As IntelliPayDotNet.Terminal = new Terminal("COM6");
terminal.TerminalResponse += New Terminal.TerminalEventHandler(terminal_TerminalResponse);
```

Define an Event Handler

(Invoke if needed)

```
C#
private void terminal_TerminalResponse(object sender, IntelliPayDotNet.TerminalEventArgs e)
{
    string rsp = e.tsdData;
    if (InvokeRequired)
    {
        BeginInvoke(new Action(() =>
        {
            txtRsp.Text = rsp;
        }));
    }
    else
    {
        txtRsp.Text = rsp;
    }
}
```

```
VB
private Sub terminal_TerminalResponse(ByVal sender As object,ByVal e As IntelliPayDotNet.TerminalEventArgs)
    DIM rsp = e.tsdData;
    if InvokeRequired Then
    {
        BeginInvoke(new Action( Function()
            txtRsp.Text = rsp;
            End Function))
    }
    Else
        txtRsp.Text = rsp;
    End If
```

SALE

```
terminal.SALE("Amount");
```

Example

```
terminal.SALE("1.00");
```

Response Example

```
{  
  "Response": {  
    "Operation": "SALE",  
    "ResponseCode": "",  
    "ResponseMsg": "SALE COMPLETED",  
    "PaymentID": "37736030",  
    "Amount": "1.00",  
    "Tip": "",  
    "Invoice": "",  
    "Clerk": "",  
    "PaymentItem": "",  
    "Card": {  
      "Entry": "CHIP",  
      "Brand": "Visa",  
      "ExpDate": "",  
      "Last4": "8491"  
    }  
  }  
}
```

VOID

```
terminal.VOID("Payment ID");
```

Example

```
terminal.VOID("37736030");
```

Response Example

```
{  
  "Response": {  
    "Operation": "VOID",  
    "ResponseCode": "",  
    "ResponseMsg": "VOID COMPLETED",  
    "PaymentID": "37736030",  
    "Amount": "",  
    "Tip": "",  
    "Invoice": "",  
    "Clerk": "",  
    "PaymentItem": ""  
  }  
}
```

REFUND

```
terminal.REFUND("Amount", "Payment ID");
```

Example

```
terminal.REFUND("1.00", "37735158");
```

Response Example

```
{  
  "Response": {  
    "Operation": "REFUN",  
    "ResponseCode": "",  
    "ResponseMsg": "REFUN COMPLETE",  
    "Payment": "37735158",  
    "Amount": "1.00",  
    "Tip": "",  
    "Invoice": "",  
    "Clerk": "",  
    "PaymentItem": ""  
  }  
}
```

ERRORS

V400

Refund Error

```
{
  "Response": {
    "Operation": "REFUN",
    "ResponseCode": "Error",
    "ResponseMsg": "Error Message",
    "PaymentID": "37735158",
    "Amount": "1.00",
    "Tip": "",
    "Invoice": "",
    "Clerk": "",
    "PaymentItem": ""
  }
}
```

Classes**Sale Return Class**

```
public class Card
{
  public string Entry { get; set; }
  public string Brand { get; set; }
  public string ExpDate { get; set; }
  public string Last4 { get; set; }
}

public class Response
{
  public string Operation { get; set; }
  public string ResponseCode { get; set; }
  public string ResponseMsg { get; set; }
  public string PaymentID { get; set; }
  public string Amount { get; set; }
  public string Tip { get; set; }
  public string Invoice { get; set; }
  public string Clerk { get; set; }
  public string PaymentItem { get; set; }
  public Card Card { get; set; }
}

public class Root
{
  public Response Response { get; set; }
}
```

VOID Class

```
public class Response
{
    public string Operation { get; set; }
    public string ResponseCode { get; set; }
    public string ResponseMsg { get; set; }
    public string PaymentID { get; set; }
    public string Amount { get; set; }
    public string Tip { get; set; }
    public string Invoice { get; set; }
    public string Clerk { get; set; }
    public string PaymentItem { get; set; }
}

public class Root
{
    public Response Response { get; set; }
}
```

REFUND Class

```
public class Response
{
    public string Operation { get; set; }
    public string ResponseCode { get; set; }
    public string ResponseMsg { get; set; }
    public string PaymentI { get; set; }
    public string Amount { get; set; }
    public string Tip { get; set; }
    public string Invoice { get; set; }
    public string Clerk { get; set; }
    public string PaymentItem { get; set; }
}

public class Root
{
    public Response Response { get; set; }
}
```

Version Log

5-14-2021/2.0.0.0

- Added Event Handlers
- Added Reconnection Handling
- When Sending Request, DLL will try to reconnect if there was an active Port at any point

5-18-2021/2.1.0.0

- Added Error Handling for the terminal when there is no carrier detected, usually meaning that the terminal is not able to send or receive from COM port.