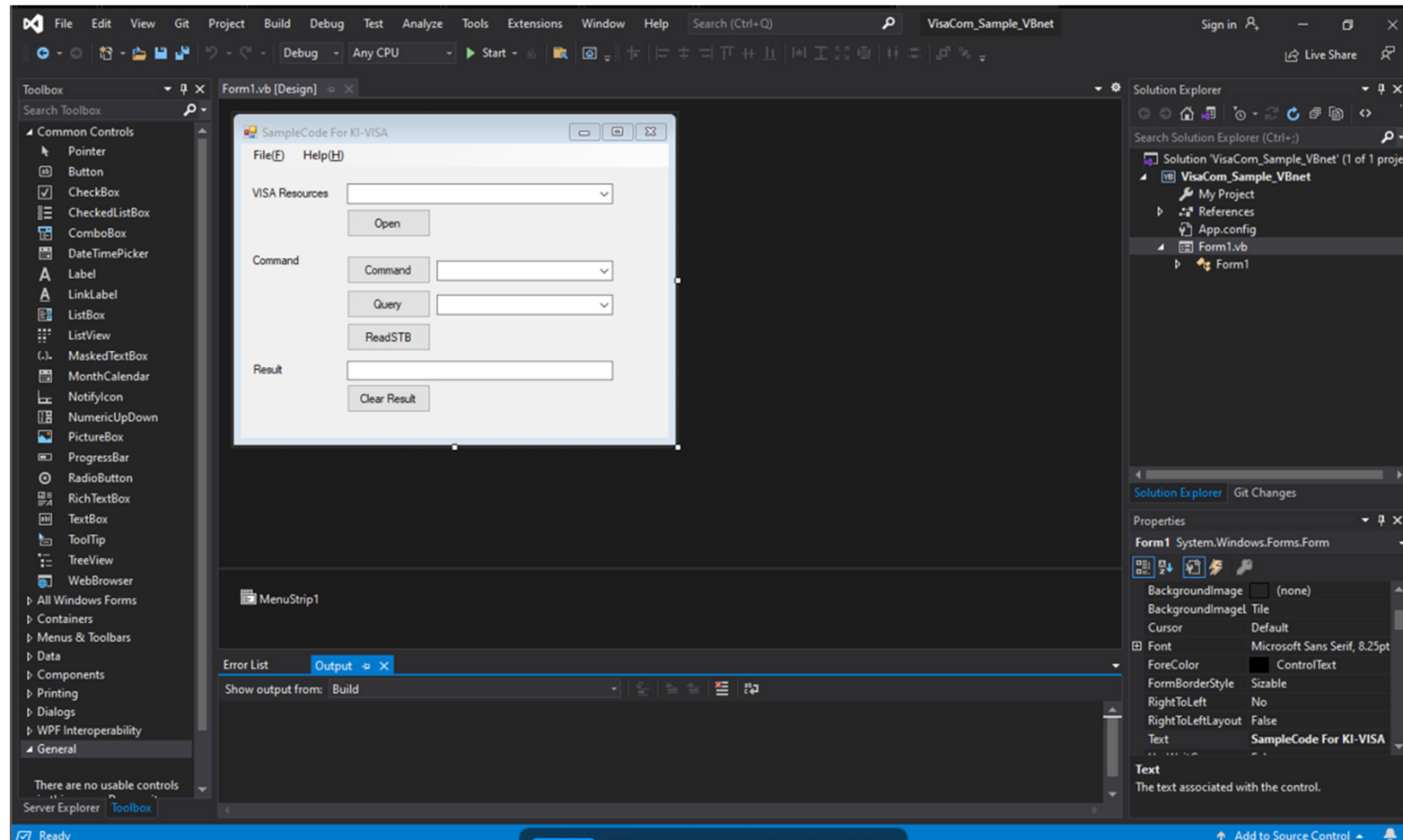


# Introduction to sample code for device control using KI-VISA library (for C#/VB.net)



# Contents

- Introduction to our sample code
- What can our sample code do?
- What is needed to run our sample code?
- Details of our sample code
- Form and Event Handler
- How to manage the reference
- How to set the target CPU
- How to solve a build error
- Guidebook / Contact Us

# Introduction to our sample code

- **Purpose of our sample code**

Our sample code simplifies the control of measurement instrumentation by utilizing the KI-VISA library for Microsoft Visual C# or Visual Basic .NET.

You can easily control devices, such as power supplies, measuring instruments and electronic loads, by modifying the initial value of our sample code, according to the communication interface manuals.

As a typical example, we provide our sample code specifically for our DC power supply PMX series.

- **Intended user**

Our sample code is designed for development engineers who need guidance in creating programming codes to control the devices using the KI-VISA library.

# What can our sample code do?

- **Connect a PC and device**

Select the VISA resource name from the VISA Resource pull-down menu. Click **Open** to open the VISA session and establish a communication link with the device.

- **Send a command**

**[Command] button**

Select the command from the Command pull-down menu. Click **Command** to send the command to the device.

- **Send a query command**

**[Query] button**

Select the command from the Query pull-down menu. Click **Query** to send the query command to the device. The result data is returned and displayed in the Result text box.

**Read a status byte register**

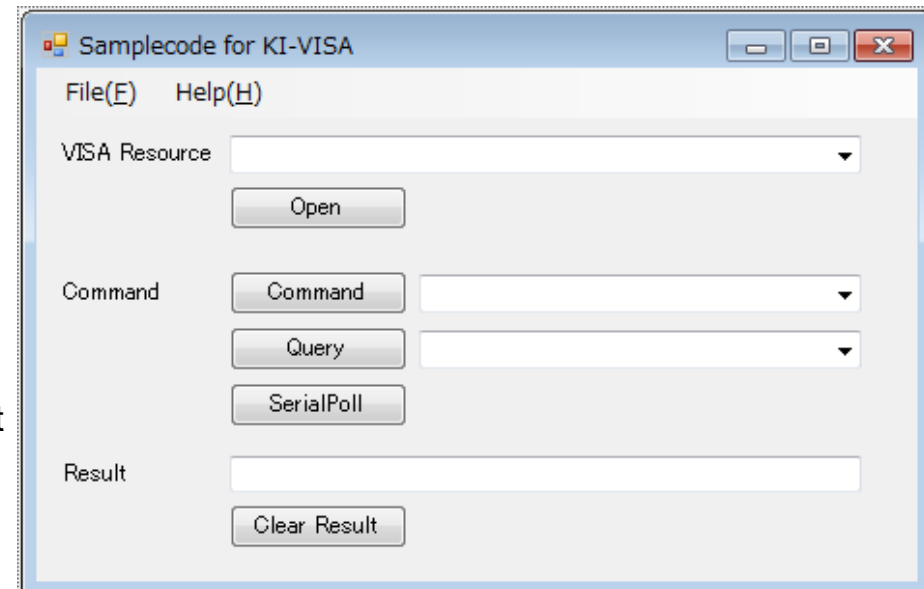
**[SerialPoll] button**

Click **SerialPoll** to execute the ReadSTB method. The value of the status byte register is displayed in the Result text box. \*Available only when connected via the GPIB or USB interface.

- **Clear the result**

**[Clear Result] button**

Clear the Result text box.



# What is needed to run our sample code?

- **Windows PC with the following requirements**

- KI-VISA library (Ver.5.5.0 or later)

<https://global.kikusui.co.jp/dri-fir-upd/ki-visa/>

\*Please note that if another VISA library, such as NI-VISA or Keysight VISA, is already installed, this library is not necessary. The reference setting is required, please see [How to manage the reference](#).

- Visual Studio 2013 or later, or Community edition

<https://visualstudio.microsoft.com/vs/community/>

\*Both editions are available if they can read the project file created with the visa library.

- **Connection cable**

You will need one of the following connection cables.

- USB connection: USB Cable (USB Type A-Type B)
- Serial connection: Serial cable (9-pin cross type)  
USB-RS232C conversion cable  
(e.g., REX-USB60F manufactured by RATOC Systems, Inc.)  
\* If your PC has a serial port, the conversion cable is not required.
- GPIB connection: USB-GPIB conversion cable  
(e.g., USB-GPIB-HS manufactured by National Instruments)  
\*Use the cable that supports the VISA COM library.  
You cannot use the GPIB-USB cable by RATOC Systems, Inc.
- LAN connection: LAN cable (cross-type)  
\*In case of AUTO-MDIX LAN interface, a straight cable is also available.

- **Device such as measuring instrument or power supply**

You will need the device with the following interfaces:

- GPIB interface
- Serial interface (RS232C)
- USB interface (USBTMC)
- LAN interface (VXI-11, HiSLIP)

# Details of our sample code

- **Contents of the ZIP file**

- The set of project including solution (.sln) file

- **Programming Language**

- Visual C# 2013 and Visual Basic 2013

\* We have verified that the file can be uploaded to Visual Studio 2019 Community edition.

- **Code Examples**

- Finding the VISA resource name of the device (FindResource method)
- How to open/close the VISA session (Open/Close method)
- Setting RS-232C baud rate
- Sending a command string (WriteString method)
- Receiving a query message (ReadString method)
- Reading the status byte by serial polling (ReadSTB method)

- **Precautions before implementing our sample code**

- You can modify our sample code, but please note that they are provided only as a reference of your programming.
- This sample code does not guarantee complete operation in all environments.
- Error handling has been minimized to simplify our sample code.  
We strongly recommend handling all errors (try-catch) on your actual system.

# Form and Event Handler

Samplecode for KI-VISA

File(E) Help(H)

VISA Resource

Open

Command

Query

SerialPoll

Result

Clear Result

```
Imports System
Imports System.IO
Imports System.Windows.Forms
Imports Ivi.Visa.Interop

'VISA library namespace specification

' Sample code for control of measuring instruments (power supply and load devices, etc.)
' using KI-VISA
' Copyright by Kikusui Electronics Corp. 2020
' Code creation: SE Section 2, Yajima.

' 2020.09.01 Ver1.00e English version(Ver1.01 based)

Public Class Form1

    'Create resource manager object
    Dim rm As IResourceManager
    Dim msg As IMessage

    'Dialog box title
    Dim sAppTitle = "KI-VISA Sample for VB.net"

    'Main form
    Public Sub New() ...

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load ...
        'Open VISA session
        Private Sub btn_Open_Click(sender As Object, e As EventArgs) Handles btn_Open.Click ...

    'Sending commands
    Private Sub btn_writeString_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btn_SendCommand.Click ...

    'Example of sending query command and receiving result
    Private Sub btn_SendQuery_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btn_SendQuery.Click ...

    'Example of serial polling the status byte register
    Private Sub btn_SerialPolling_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btn_SerialPolling.Click ...

    'Processing when the form was closed
    Private Sub quitToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles QuitToolStripMenuItem.Click ...

    'Clear the Result text box
    Private Sub btn_ClearResult_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btn_ClearResult.Click ...

    'AboutBox Version/Contact Information Display
    Private Sub aboutToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles AboutToolStripMenuItem.Click ...

End Class
```

# How to manage the reference

If you want to create a new project or [are experiencing issues with building a project correctly](#), please refer to the following steps:

- Check **VISA COM 5.x Type Library** and click **OK**. \* The "x" in "5.x" depends on your VISA version.
- If the VISA COM 5.x Type Library is not found, add a reference based on your VISA version:

For 32-bit OS:

C:\Program Files\IVI Foundation\VISA\VisaCom\GlobMgr.dll

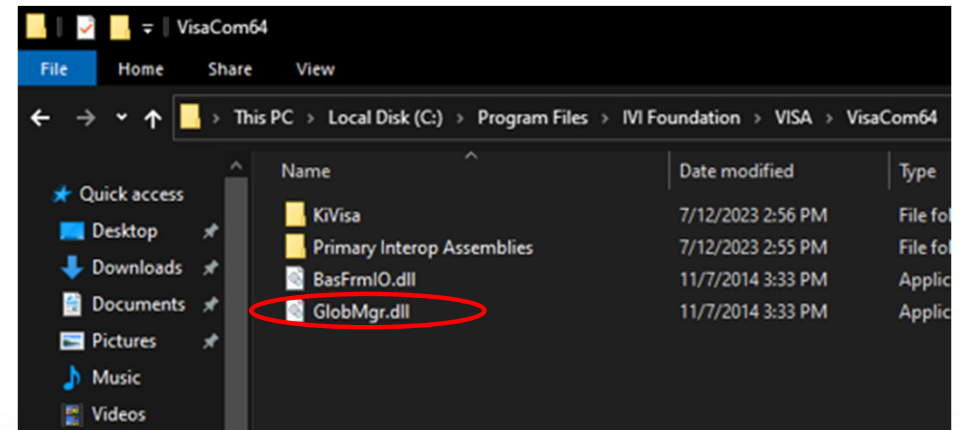
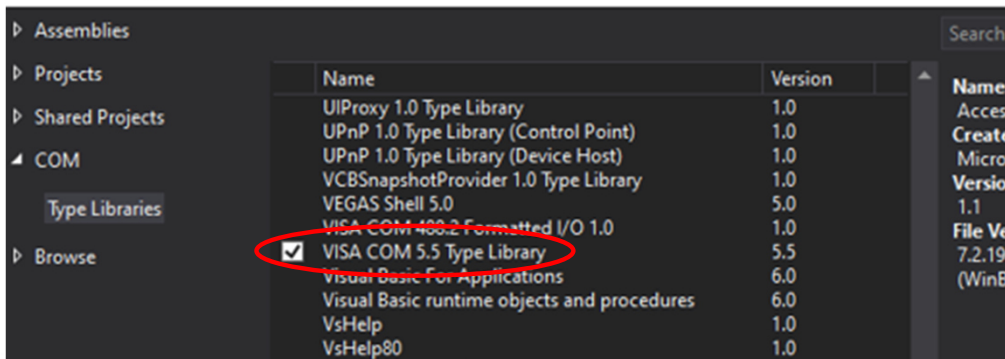
For 64-bit OS:

32-bit (x86): C:\Program Files (x86)\IVI Foundation\VISA\VisaCom\GlobMgr.dll

or

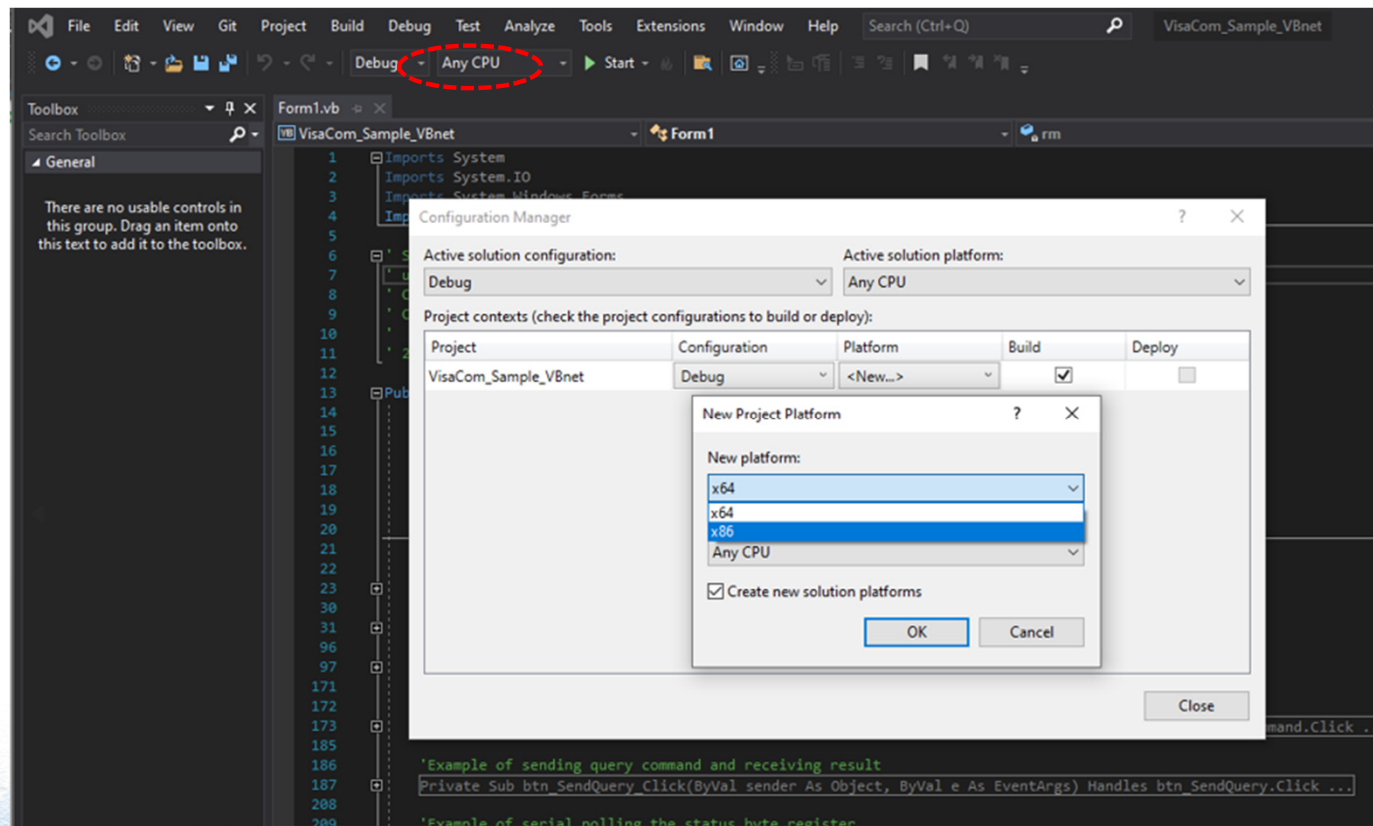
64-bit (x64): C:\Program Files\IVI Foundation\VISA\VisaCom64\GlobMgr.dll

Reference Manager - VisaCom\_Sample\_VBnet



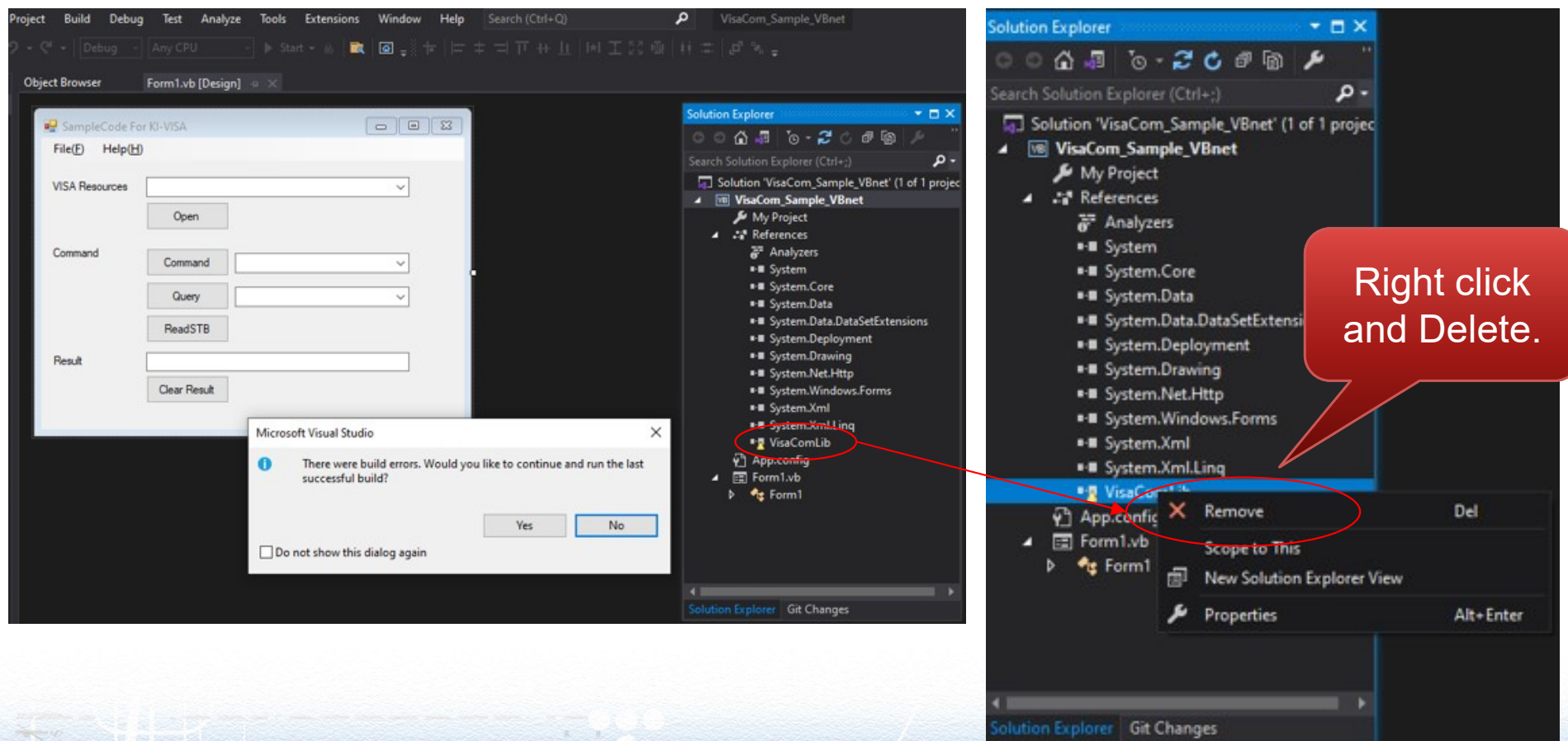
# How to set the target CPU

- To set the target CPU for your project, follow these steps:  
Open the Configuration Manager and select **Any CPU**. This ensures that our sample code will run correctly regardless of the specific CPU architecture. If you need to use a driver specifically designed for a 32-bit environment, choose **x86** (32-bit build).
- When selecting x86 as the project platform, if the VISA COM Type Library of the 64-bit (x64) version is referenced in Visual Studio, the program will automatically reference the 32-bit version driver.



# How to solve a build error

- If your program is not properly referencing the VISA COM library and you encounter a build error, you can follow these steps to resolve the issue: go to View > Solution Explorer to delete the existing VISA COM library reference and [re-install it again](#).



# Guidebook / Contact Us

- Please download the KI-VISA programming guidebook for reference (Note: The guidebook is available in Japanese text only):

<https://kikusui.co.jp/kikusupport/downloads/guidebook/programing-guidebook/>

- For any inquiries regarding the sample code, please contact us using the following inquiry form:

<https://global.kikusui.co.jp/contact/lead/>

Please include the phrase “To the person in charge of the sample code” in the text and provide your question.

```

1 Imports System
2 Imports System.IO
3 Imports System.Windows.Forms
4 Imports Ivi.Visa.Interop 'VISA library namespace specification
5
6 ' Sample code for control of measuring instruments (power supply and load devices, etc.)
7 ' using KI-VISA
8 ' Copyright by Kikusui Electronics Corp. 2020
9 ' Code creation: SE Section 2, Yajima.
10
11 ' 2020.09.01 Ver1.00e English version(Ver1.01 based)
12
13 Public Class Form1
14
15     ' Create resource manager object
16     Dim rm As IResourceManager3
17     Dim msg As IMessage
18
19     ' Dialog box title
20     Dim sAppTitle = "KI-VISA Sample for VB.net"
21
22     ' Main form
23     Public Sub New()
24         InitializeComponent()
25
26         ' Specify form display position in screen center
27         Me.StartPosition = FormStartPosition.CenterScreen
28
29     End Sub
30
31     Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
32
33         ' Storage variable of VISA resource list found by FindResource
34         Dim foundResources() As String
35
36         ' Initialize the Resource Manager
37         rm = New ResourceManager()
38
39         ' Get the name of a VISA resource containing an INSTR available in FindResource
40         ' For more information, please refer to KI-VISA Library VISA COM Guidebook 8 - Dynamically Setting
41         ' VISA Addresses (starting on page 48).
42         Try
43             foundResources = rm.FindRsrc("?*INSTR")
44
45         Catch ex As Exception
46             MessageBox.Show _
47                 ("No available VISA resources found on the computer." _
48                 + vbCrLf + "Make sure your device is connected",
49                 sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop)
50             Exit Sub
51         End Try
52
53         ' Add the acquired VISA resource name to the VISA Resource Name combo box.
54         cmb_Resources.DataSource = foundResources
55
56         ' Add a setting command to the combo box.
57         ' Commands not listed in the combo box can be entered manually at run time.
58         ' If you edit the commands here, you can select any command from the pull-down menu.
59         ' As a typical example, the following commands are available for the
60         ' PMX18-5A (DC power supply 18V/5A model).
61         cmb_CommandString.Items.Add("SYST:REMOTE")
62         cmb_CommandString.Items.Add("SYST:LOCAL")
63         cmb_CommandString.Items.Add("*CLS")
64         cmb_CommandString.Items.Add("OUTP ON")
65         cmb_CommandString.Items.Add("OUTP OFF")
66         cmb_CommandString.Items.Add("VOLT 10")
67         cmb_CommandString.Items.Add("VOLT 1")
68         cmb_CommandString.Items.Add("CURR 1")
69         cmb_CommandString.Items.Add("CURR 0.1")
70         cmb_CommandString.SelectedIndex = 0 'Display the first command

```

```

72
73 ' Add a query command to the combo box
74 cmb_QueryString.Items.Add("*IDN?")
75 cmb_QueryString.Items.Add("*STB?")
76 cmb_QueryString.Items.Add("SYST:ERR?")
77 cmb_QueryString.Items.Add("OUTP?")
78 cmb_QueryString.Items.Add("VOLT?")
79 cmb_QueryString.Items.Add("CURR?")
80 cmb_QueryString.Items.Add("READ:VOLT?")
81 cmb_QueryString.Items.Add("READ:CURR?")
82 cmb_QueryString.SelectedIndex = 0 ' Show the first query command
83
84 ' Disable button before opening a VISA session.
85 btn_SendCommand.Enabled = False
86 btn_SendQuery.Enabled = False
87 btn_SerialPolling.Enabled = False
88
89 ' -----
90 ' [Caution] Searching for LAN Devices
91 ' Dynamic search may not be available for LAN devices.
92 ' In that case, you can use it by Finding with KI-VISA Instruments Explorer.
93 ' -----
94
95 End Sub
96 ' Open VISA session
97 Private Sub btn_Open_Click(sender As Object, e As EventArgs) Handles btn_Open.Click
98 ' For more information, see KI-VISA Library VISA COM Guidebook 6 – Opening VISA Sessions (starting
on page 35).
99 Dim gpib As IGpib
100 Dim seri As ISerial
101 Dim tcp As ITcpipInstr
102 Dim usb As IUsb
103
104 ' Open a VISA session with the resource name specified in the combo box
105 Try
106 msg = rm.Open(cmb_Resources.Text, AccessMode.NO_LOCK, 0, "") ' Support the change of timeout
specification in NI-VISA
107 MessageBox.Show("VISA session opened successfully!" + vbCrLf +
108 cmb_Resources.Text, sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Asterisk)
109
110 Catch ex As Exception
111 MessageBox.Show("VISA session opening failed." + vbCrLf +
112 cmb_Resources.Text, sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop)
113 Exit Sub
114
115 End Try
116
117 ' Configuration by Interface Type
118 ' Please refer to KI-VISA Library VISA COM Guidebook 7 – Basic IO (from page 39 onwards) for details.
119
120 msg.TerminationCharacter = 10
121 msg.TerminationCharacterEnabled = True ' When performing ReadString() on an IEEE 488.2 measuring
instrument, ' "true" setting is recommended for correct terminating.
122 msg.SendEndEnabled = True
123
124 ' Configure interface type by HardwareInterfaceType
125 Select Case msg.HardwareInterfaceType
126 Case 1
127 ' GPIB
128 gpib = msg
129 gpib.RepeatAddressingEnabled = True
130 gpib.UnaddressingEnabled = False
131
132 ' Enable the serial polling button (since only GPIB and USB can be used)
133 btn_SerialPolling.Enabled = True
134
135 Case 4
136 ' ASRL (Serial port)
137 ' You can set the baud rate by changing the parameters here.
138 seri = msg
139 seri.BaudRate = 19200 ' baud rate 19200bps

```

```

140         seri.DataBits = 8                                'Data Bits 8bit
141         seri.StopBits = SerialStopBits.ASRL_STOP_ONE    'Stop Bits 1bit
142         seri.Parity = SerialParity.ASRL_PAR_NONE        'No parity
143         seri.FlowControl = SerialFlowControl.ASRL_FLOW_NONE 'Without flow control
144
145         'Disabling the serial polling button
146         btn_SerialPolling.Enabled = False
147
148     Case 6
149         'LAN
150         '[Caution] Dynamic search may not be available for LAN devices.
151         'In that case, you can use it by Finding with KI-VISA Instruments Explorer.
152         tcp = msg
153
154         'Disabling the serial polling button
155         btn_SerialPolling.Enabled = False
156
157     Case 7
158         'USB
159         usb = msg
160
161         'Enable the serial polling button(since only GPIB and USB can be used)
162         btn_SerialPolling.Enabled = True
163
164     End Select
165
166     'Enable the button.
167     btn_SendCommand.Enabled = True
168     btn_SendQuery.Enabled = True
169
170 End Sub
171
172 'Sending commands
173 Private Sub btn_writeString_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btn_SendCommand.
Click
174     Dim r As Integer
175
176     'Send the command selected by cmb_CommandString
177     r = msg.WriteString(cmb_CommandString.Text + vbCrLf) 'Add a delimiter after the command message
178
179
180     'If you want to write the command directly, you can also write it like this.
181     'r = msg.WriteString("VOLT 10" + vbCrLf)
182     'r = msg.WriteString("OUTP ON" + vbCrLf)
183
184 End Sub
185
186 'Example of sending query command and receiving result
187 Private Sub btn_SendQuery_Click(ByVal sender As Object, ByVal e As EventArgs) Handles btn_SendQuery.
Click
188     Dim r As Integer
189     Dim strret As String = ""
190
191     'query command transmission
192     r = msg.WriteString(cmb_QueryString.Text + vbCrLf) 'Add a delimiter after the command message
193
194
195     'Receive query results
196     Try
197         strret = msg.ReadString(1024) 'Buffer size 1024 bytes
198
199     Catch ex As Exception
200         MessageBox.Show("Failed to receive query message.",
201             sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop)
202     End Try
203
204     'Returns the value received by ReadString.
205     tb_Result.Text = strret
206 End Sub
207
208 'Example of serial polling the status byte register
209

```

Form1.vb

```
210 Private Sub btn_SerialPolling_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
    btn_SerialPolling.Click
211
212     Dim stb As Short
213
214     'Serial polling the status byte register.
215     'ReadSTB is only available for GPIB and USB interfaces.
216     'For other interfaces, the button are disabled. (because exception is occur)
217     stb = msg.ReadSTB()
218
219     'Write result data to text box
220     tb_Result.Text = stb.ToString()
221 End Sub
222
223 'Processing when the form was closed
224 Private Sub quitToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
    QuitToolStripMenuItem.Click
225     'Close the VISA session
226     If Not msg Is Nothing Then
227         msg.Close()
228     End If
229     'Close the form
230     Me.Close()
231 End Sub
232
233 'Clear the Result text box
234 Private Sub btn_ClearResult_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
    btn_ClearResult.Click
235
236     'Clear the Result text box
237     tb_Result.Text = ""
238 End Sub
239
240 'AboutBox Version/Contact Information Display
241 Private Sub aboutToolStripMenuItem_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
    AboutToolStripMenuItem.Click
242
243     MessageBox.Show(
244         "Samplecode for KI-VISA for VB.net Ver1.00e" + vbCrLf +
245         "Copyright by Kikusui Electronics Corp. 2020" + vbCrLf + sAppTitle)
246
247 End Sub
248
249
250 End Class
```

```

1 using System;
2 using System.IO;
3 using System.Windows.Forms;
4 using Ivi.Visa.Interop;           // VISA library namespace specification
5
6 // Sample code for measuring instrument (power supply, load device, etc.) using KI-VISA
7 // Copyright by Kikusui Electronics Corp. 2015-2020
8 // Code Creation: SE2 Section Yajima
9
10 // 2020.09.01 Ver1.00e (Ver1.05 base)
11
12
13 namespace SampleCode_KIVISA
14 {
15     public partial class Form1 : Form
16     {
17         // Create resource manager object
18         IResourceManager3 rm;
19         IMessage msg;
20
21         // Dialog box title
22         readonly string sAppTitle = "KI-VISA Sample for CSharp";
23
24         // Main form
25         public Form1()
26         {
27             InitializeComponent();
28             //Set the form display position to the center of the screen.
29             this.StartPosition = FormStartPosition.CenterScreen;
30         }
31
32         // Processing when displaying form
33         private void Form1_Load(object sender, EventArgs e)
34         {
35
36             // A storage variable for the VISA resource list found by FindResource
37             string[] foundResources;
38
39             // Initialize the Resource Manager
40             rm = new ResourceManager();
41
42             // Get the name of a VISA resource containing an INSTR available in FindResource
43             // For more information, please refer to KI-VISA Library
44             // VISA COM Guidebook 8 - Dynamically Setting VISA Addresses (starting on page 48).
45             try
46             {
47                 foundResources = rm.FindRsrc("?*INSTR");
48             }
49             catch (Exception)
50             {
51                 MessageBox.Show("No available VISA resources found on the computer." + "¥r"
52                     + "Make sure your device is connected",
53                     sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop);
54                 return;
55             }
56
57
58             // Add the acquired VISA resource name to the VISA Resource Name combo box.
59             cmb_Resources.DataSource = foundResources;
60
61
62             // Add a setting command to the combo box.
63             // Commands not listed in the combo box can be entered manually at run time.
64             // If you edit the commands here, you can select any command from the pull-down menu.
65             // As a typical example, the following commands are available for the
66             // PMX18-5A (DC power supply 18V/5A model).
67             cmb_CommandString.Items.Add("SYST:REMOTE");
68             cmb_CommandString.Items.Add("SYST:LOCAL");
69             cmb_CommandString.Items.Add("*CLS");
70             cmb_CommandString.Items.Add("OUTP ON");
71             cmb_CommandString.Items.Add("OUTP OFF");
72             cmb_CommandString.Items.Add("VOLT 10");

```

```

73     cmb_CommandString.Items.Add("VOLT 1");
74     cmb_CommandString.Items.Add("CURR 1");
75     cmb_CommandString.Items.Add("CURR 0.1");
76     cmb_CommandString.SelectedIndex = 0;           // Display the first command
77
78     // Add a query command to the combo box
79     cmb_QueryString.Items.Add("*IDN?");
80     cmb_QueryString.Items.Add("*STB?");
81     cmb_QueryString.Items.Add("SYST:ERR?");
82     cmb_QueryString.Items.Add("OUTP?");
83     cmb_QueryString.Items.Add("VOLT?");
84     cmb_QueryString.Items.Add("CURR?");
85     cmb_QueryString.Items.Add("READ:VOLT?");
86     cmb_QueryString.Items.Add("READ:CURR?");
87     cmb_QueryString.SelectedIndex = 0;           // Display the first command
88
89     // Disable button before opening a VISA session.
90     btn_SendCommand.Enabled = false;
91     btn_SendQuery.Enabled = false;
92     btn_SerialPolling.Enabled = false;
93
94     //-----
95     // [Caution] Searching for LAN Devices
96     // Dynamic search may not be available for LAN devices.
97     // In that case, you can use it by Finding with KI-VISA Instruments Explorer.
98     //-----
99 }
100
101 // Open VISA session
102 private void btn_Open_Click(object sender, EventArgs e)
103 {
104     // For more information, see KI-VISA Library VISA COM Guidebook 6 – Opening VISA Sessions (
starting on page 35).
105     IGpib gpib;
106     ISerial seri;
107     ITcpipInstr tcp;
108     IUsb usb;
109
110     // Open a VISA session with the resource name specified in the combo box
111     try
112     {
113         msg = (IMessage)rm.Open(cmb_Resources.Text, AccessMode.NO_LOCK, 0, ""); // Support the
change of timeout specification in NI-VISA
114         MessageBox.Show("VISA session opened successfully!" + "¥r" + cmb_Resources.Text, sAppTitle,
MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
115     }
116     catch (Exception)
117     {
118         MessageBox.Show("VISA session opening failed." + "¥r" + cmb_Resources.Text, sAppTitle,
MessageBoxButtons.OK, MessageBoxIcon.Stop);
119         return;
120     }
121
122     // Configuration by Interface Type
123     // Please refer to KI-VISA Library VISA COM Guidebook 7 – Basic IO (from page 39 onwards) for
details.
124     msg.TerminationCharacter = 10;
125     msg.TerminationCharacterEnabled = true; // When performing ReadString() on an IEEE 488.2
measuring instrument,
126                                           // "true" setting is recommended for correct terminating.
127     msg.SendEndEnabled = true;
128
129     // Configure interface type by HardwareInterfaceType
130     switch (msg.HardwareInterfaceType)
131     {
132     case 1:
133         // GPIB
134         gpib = (IGpib)msg;
135         gpib.RepeatAddressingEnabled = true;
136         gpib.UnaddressingEnabled = false;
137
138         // Enable the serial polling button (since only GPIB and USB can be used)

```

```

139         btn_SerialPolling.Enabled = true;
140
141         break;
142
143     case 4:
144         // ASRL (Serial port)
145         // You can set the baud rate by changing the parameters here.
146         seri = (ISerial)msg;
147         seri.BaudRate = 19200; // baud rate 19200bps
148         seri.DataBits = 8; // Data Bits 8bit
149         seri.StopBits = SerialStopBits.ASRL_STOP_ONE; // Stop Bits 1bit
150         seri.Parity = SerialParity.ASRL_PAR_NONE; // No parity
151         seri.FlowControl = SerialFlowControl.ASRL_FLOW_NONE; // Without flow control
152
153         // Disabling the serial polling button
154         btn_SerialPolling.Enabled = false;
155
156         break;
157
158     case 6:
159         // LAN
160         // [Caution] Dynamic search may not be available for LAN devices.
161         // In that case, you can use it by Finding with KI-VISA Instruments Explorer.
162         tcp = (ITcpipInstr)msg;
163
164         // Disabling the serial polling button
165         btn_SerialPolling.Enabled = false;
166
167         break;
168
169     case 7:
170         // USB
171         usb = (IUsb)msg;
172
173         // Enable the serial polling button (since only GPIB and USB can be used)
174         btn_SerialPolling.Enabled = true;
175
176         break;
177     }
178
179     // Enable the button.
180     btn_SendCommand.Enabled = true;
181     btn_SendQuery.Enabled = true;
182
183 }
184
185 // Sending commands
186 private void btn_writeString_Click(object sender, EventArgs e)
187 {
188     int r;
189
190     // Send the command selected by cmb_CommandString
191     r = msg.WriteString(cmb_CommandString.Text + "\n"); // Add a delimiter after the command
192
193     // If you want to write the command directly, you can also write it like this.
194     // r = msg.WriteString("VOLT 10\n");
195     // r = msg.WriteString("OUTP ON\n");
196 }
197
198 // Example of sending query command and receiving result
199 private void btn_SendQuery_Click(object sender, EventArgs e)
200 {
201     int r;
202     string strret = "";
203
204     // query command sending
205     r = msg.WriteString(cmb_QueryString.Text + "\n"); // Add a delimiter after the command
206
207     // Receive query results
208     try

```

Form1.cs

```
209     {
210         strret = msg.ReadString(1024); // Buffer size 1024 bytes
211     }
212     catch (Exception)
213     {
214         MessageBox.Show("Failed to receive query message."
215             + "¥r" + cmb_Resources.Text,
216             sAppTitle, MessageBoxButtons.OK, MessageBoxIcon.Stop);
217         return;
218     }
219
220
221     //Returns the value received by ReadString.
222     tb_Result.Text = strret;
223 }
224
225 //Example of serial polling the status byte register
226 private void btn_SerialPolling_Click(object sender, EventArgs e)
227 {
228     short stb;
229
230     // Serial polling the status byte register.
231     // ReadSTB is only available for GPIB and USB interfaces.
232     // For other interfaces, the button are disabled. (because exception is occur)
233     stb = msg.ReadSTB();
234
235     // Write result data to text box
236     tb_Result.Text = stb.ToString();
237 }
238
239 // Processing when the form was closed
240 private void quitToolStripMenuItem_Click(object sender, EventArgs e)
241 {
242     // Close the VISA session
243     if (msg != null)
244     {
245         msg.Close();
246     }
247     // Close the form
248     this.Close();
249 }
250
251 // Clear the Result text box
252 private void btn_ClearResult_Click(object sender, EventArgs e)
253 {
254     // Clear the Result text box
255     tb_Result.Text = "";
256 }
257
258 // AboutBox Version/Contact Information Display
259 private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
260 {
261     MessageBox.Show(
262         "Samplecode for KI-VISA CSharp Ver1.00e" + "¥r"
263         + "Copyright by Kikusui Electronics Corp. 2015-2020" + "¥r", sAppTitle);
264 }
265 }
266 }
267
268
269 }
```