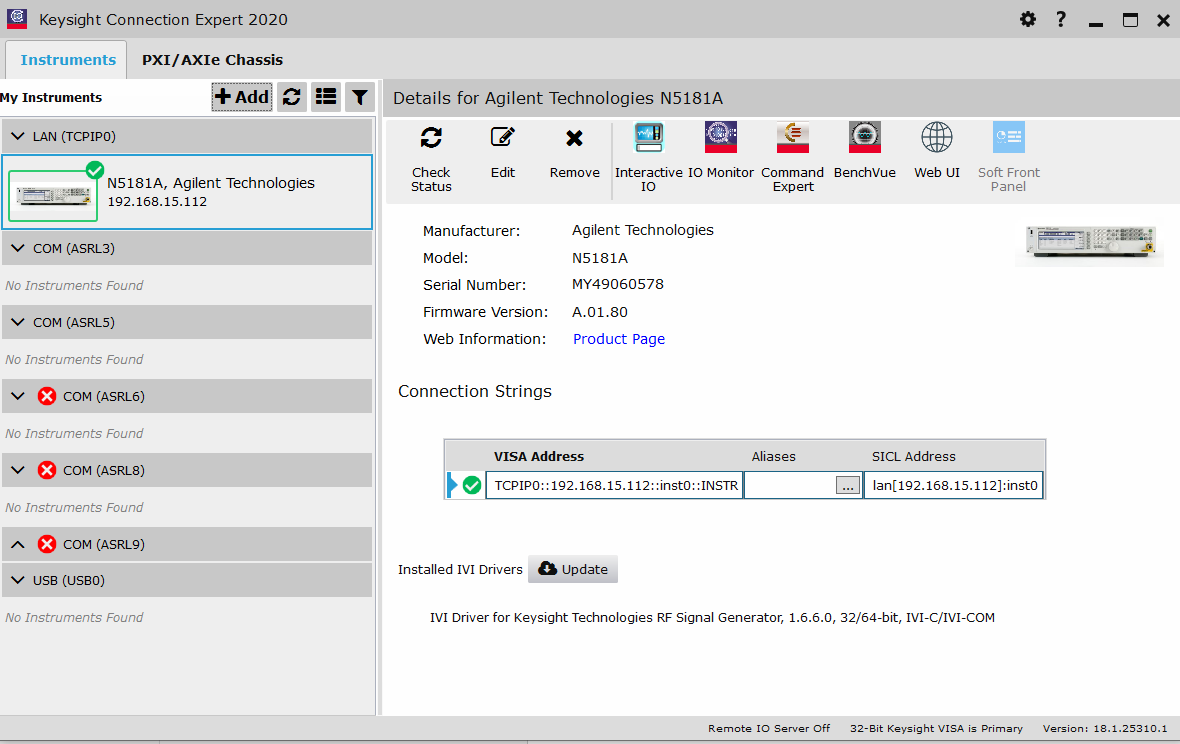
Automatic detect instrument

There is a C# way of automatic detect the instrument

We will show an example with the N5181A – Signal Generator

Open Keysight Io Control

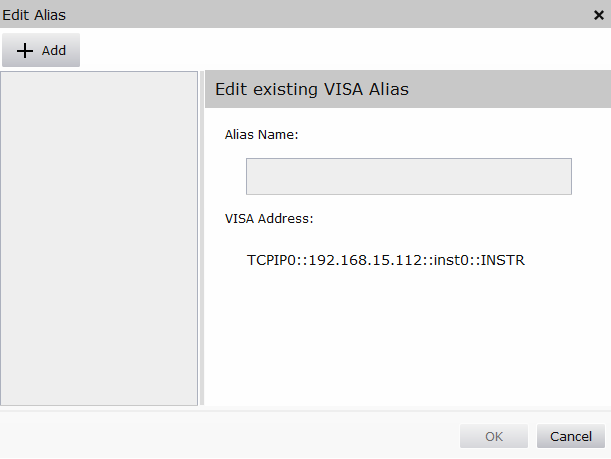


Scan to detect your instrument

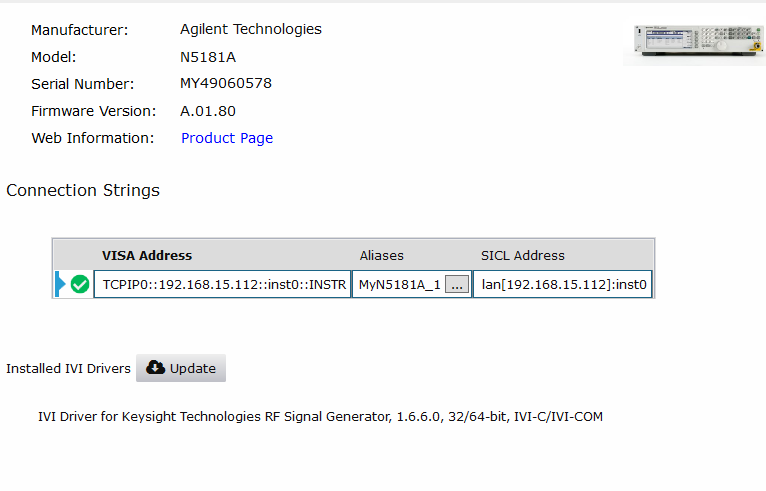
At first time when you detect it , the Alias will be empty

Press the three dots…

And give the instrument the Alias you want



If you have two of a kind you can give MyN5181A\_1 and MyN5181A\_2 to identify them in the software



I named it here MyN5181A\_1

Open Find devices project

Code is attached here:

////////////////////////////////////////////////////////////////////////////////

// © Keysight Technologies 2016

//

// You have a royalty-free right to use, modify, reproduce and distribute

// the Sample Application Files (and/or any modified version) in any way

// you find useful, provided that you agree that Keysight Technologies has no

// warranty, obligations or liability for any Sample Application Files.

//

////////////////////////////////////////////////////////////////////////////////

using System;

using System.Collections.Generic;

using System.Text;

using Ivi.Visa;

using Ivi.Visa.FormattedIO;

namespace FindDevices

{

class Program

{

static void Main(string[] args)

{

IEnumerable<string> devices;

try

{

// Finding all devices and interfaces is straightforward

Console.WriteLine("Find all devices and interfaces:");

devices = GlobalResourceManager.Find();

foreach (string device in devices)

{

Console.WriteLine("\tAddress: {0}, Alias: {1}", device, GlobalResourceManager.Parse(device).AliasIfExists);

}

}

catch (VisaException ex)

{

Console.WriteLine("Didn't find any devices!");

}

Console.WriteLine();

// You can specify other device types using different search strings. Here are some common examples:

// All instruments (no INTFC, BACKPLANE or MEMACC)

Find("?\*INSTR");

// PXI modules

Find("PXI?\*INSTR");

// USB devices

Find("USB?\*INSTR");

// GPIB instruments

Find("GPIB?\*");

// GPIB interfaces

Find("GPIB?\*INTFC");

// GPIB instruments on the GPIB0 interface

Find("GPIB0?\*INSTR");

// LAN instruments

Find("TCPIP?\*");

// SOCKET (::SOCKET) instruments

Find("TCPIP?\*SOCKET");

// VXI-11 (inst) instruments

Find("TCPIP?\*inst?\*INSTR");

// HiSLIP (hislip) instruments

Find("TCPIP?\*hislip?\*INSTR");

// RS-232 instruments

Find("ASRL?\*INSTR");

Console.WriteLine("Press any key to exit...");

Console.ReadKey();

}

static void Find(string searchString)

{

IEnumerable<string> devices;

try

{

Console.WriteLine("Find with search string \"" + searchString + "\"");

devices = GlobalResourceManager.Find(searchString);

foreach (string device in devices)

{

Console.WriteLine("\tAddress: {0}, Alias: {1}", device, GlobalResourceManager.Parse(device).AliasIfExists);

}

}

catch (VisaException ex)

{

Console.WriteLine("... didn't find anything!");

}

Console.WriteLine();

}

}

}

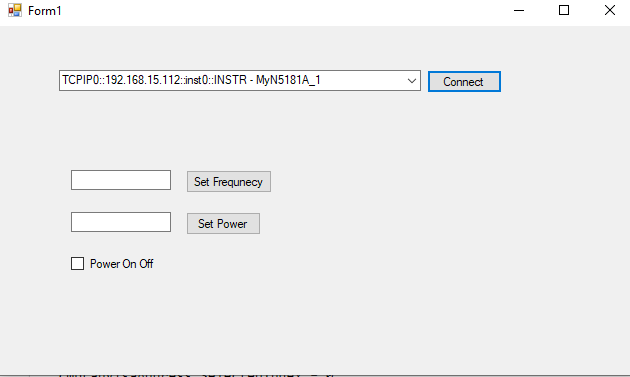
You already know where you connected the instrument, if its USB, TCPIP ( LAN ) Socket and so on

The result of the find devices show you your device and its Alias.



You can now take the visa address and use it in your code.

I Created an API from this code:



if (KeysightFindDevices.FindDevices(INTERFACE\_TYPE.TCPIP\_LAN, out List<KeysightDevice> deviceList, out string outMessage) == false)

{

MessageBox.Show("Failed to find devices: " + Environment.NewLine + outMessage);

return;

} else

{

if (deviceList.Count > 0)

{

foreach (KeysightDevice d in deviceList)

{

cmbLanVisaAddress.Items.Add(d.VisaAddress + " - " + d.Alias);

}

cmbLanVisaAddress.SelectedIndex = 0;

}

}

The function search for one interface and return a list of the devices in that interface

If you have devices from several interfaces ( USB , LAN) , call the function twice and holds two lists of a kind.