# Getting started with the MegiQ VNA PI in C#

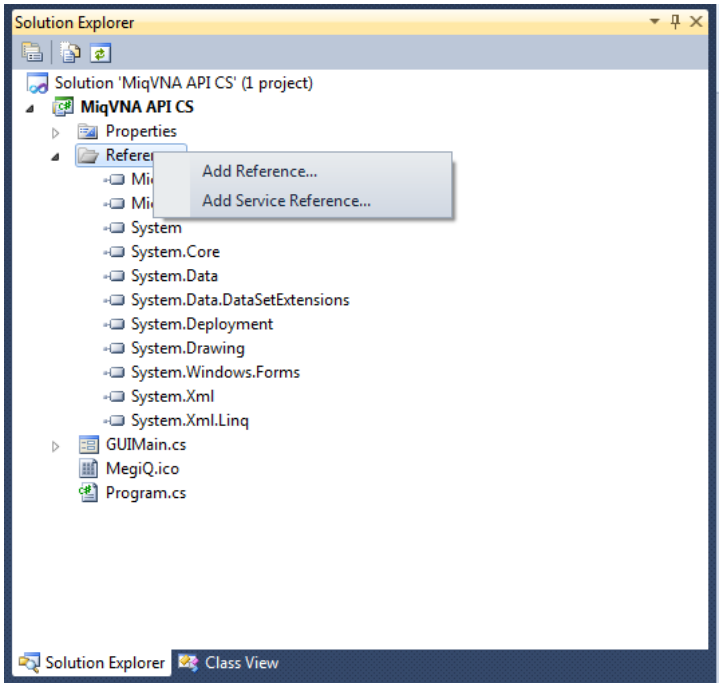
## 1 Introduction

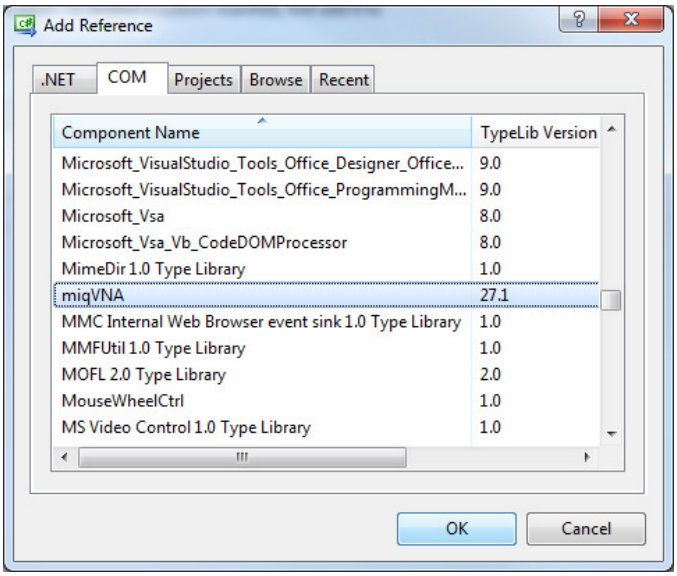
This document shows how to create a new VNA project in C#.

A VNA project must include a reference to the VNA API library so that the compiler can reference the entities in the library.

## 2 Creating a new Project

- Make sure the MegiQ VNA program is installed.
- Create a new project
- In the Solution Explorer, right click on References and select Add Reference…




- In the COM tab, find miqVNA and click OK.
- In your .cs file, add the following:

```
using miqVNA;
```

## 3 Connecting to the VNA

Connecting to MiQVNA is as easy at this:

```
private mvnaVNAMain myVNA;
myVNA = new mvnaVNAMain();
```

This will cause the MiQVNA program to be started and connected to your application. From this object all properties and methods as described in the MegiQ VNA Application Programming Interface can be used.

The MiQVNA program also generates events. The following sample shows how to add an event handler.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using miqVNA;

namespace MyFirstVNA
{
        private mvnaVNAMain myVNA;

        public Form1()
        {
            InitializeComponent();
            myVNA = new mvnaVNAMain();

            myVNA.evtSweepProgress += new __mvnaVNAMain_evtSweepProgressEventHandler(myVNA_evtSweepProgress);
        }

        void myVNA_evtSweepProgress(int PointsReceived, int PointsTotal)
        {
            // GUI operations here cause threading issues
            progressBar1.Maximum = PointsTotal;  // ← ERROR!
            progressBar1.Value = PointsReceived;
        }
    }
}
```

This will cause a run time error, because the events are coming from another thread than the one that created the GUI (controls).

The use of “InvokeRequired” and delegates can be used to solve this issue:

```
        void myVNA_evtSweepProgress(int PointsReceived, int PointsTotal)
        {
            // GUI operations here cause threading issues
            if (progressBar1.InvokeRequired)
            {
                progressBar1.BeginInvoke(new MethodInvoker(delegate()
                {
                    myVNA_evtSweepProgress(PointsReceived, PointsTotal);
                }));
            }
            else
            {
                progressBar1.Maximum = PointsTotal;
                progressBar1.Value = PointsReceived;
            }
        }
```

# 4 And further

You can find many functions and procedures for controlling the VNA, acquiring data and plotting results in the example C# project.