

# NC Viewer .Net Control



Making Graphics in the Real World Simple...

## PRODUCT DESCRIPTION:

This is a .NET control designed to be embedded into your custom machine automation HMI.

Use with Visual Studio WinForm and WPF projects in VB or C# to display DXF files and/or NC type files which represent some motion or tool path being executed in real-time.

Drop this control on your HMI Panel or Dialog then create buttons, sliders, checkboxes as needed to control the many features of the NC-Viewer control.

## FEATURES :

- Load DXF or NC files or both (overlay).
- Can customize the NC (G-Code) reader for your specific machine. (3 hours included)
- Fully customize colors and line types for different NC entities. Rapid different from G01...etc.
- Colors and Line types can change as NC code is executed in real time with simple update of properties.
- Create a custom Tool that moves with real time updates along the Tool Path (can show Angle as well).
- Makes it easy to load and display a 2D tool path in your HMI. (used extensively now with PMAC systems)
- Full working examples w/ source (not Control source) in VB and C#.
- Allows for easy Tool Path Simulation (cutting) in the display with no machine movement.
- Easily synchronized with NC commands as executed in a typical list box (control is aware of NC line numbers).

## ORDERING & OPTIONS:

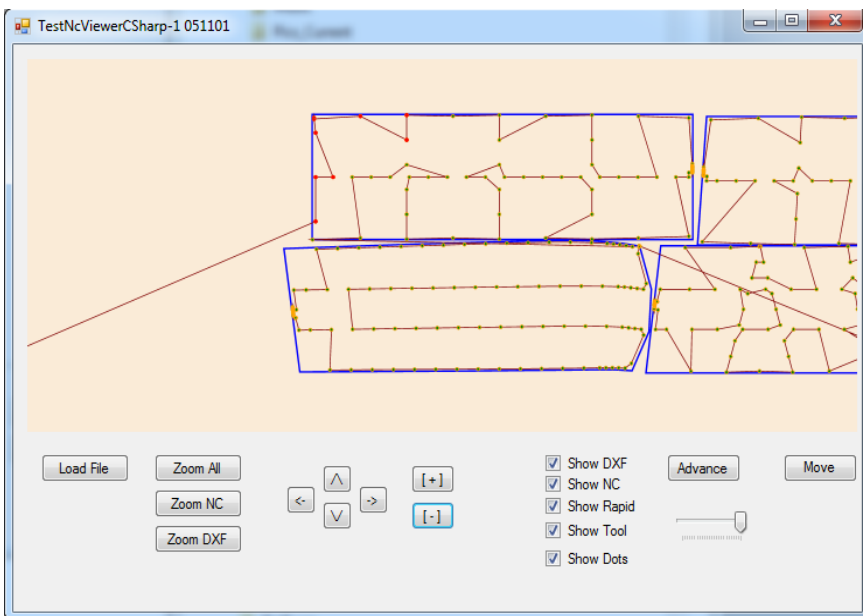
NC Viewer .NET Control                      \$2250.00  
-includes up to 3 hours of customization  
-includes VB and C# Visual Studio .Net examples

Extra Customization work                      \$ 225.00/hr.  
-only if needed for very complex or unique NCfiles

*Royalty Free distribution inside your custom application.  
You are Not allowed to distribute or re-sell the Control stand-alone. Control source code is NOT provided.*

## Information & Ordering:

[www.TotalLLC.com](http://www.TotalLLC.com)    [sales@TotalLLC.com](mailto:sales@TotalLLC.com)



```
//NcNote1 - here add reference to using the control in your application using GCodeViewer;

namespace TestNcViewer_CSharp_1
{
    public partial class Form1 : Form
    {
        GCodeView viewer = new GCodeView();

        public Form1()
        {
            InitializeComponent();
            this.elementHost1.Child = viewer;
        }...

        this.viewer.CurrentNcBlock++;
        this.viewer.ToolAngle = this.ToolAngle.Value;
    }
}
```