



White Paper

HTML5-Remoting solution for

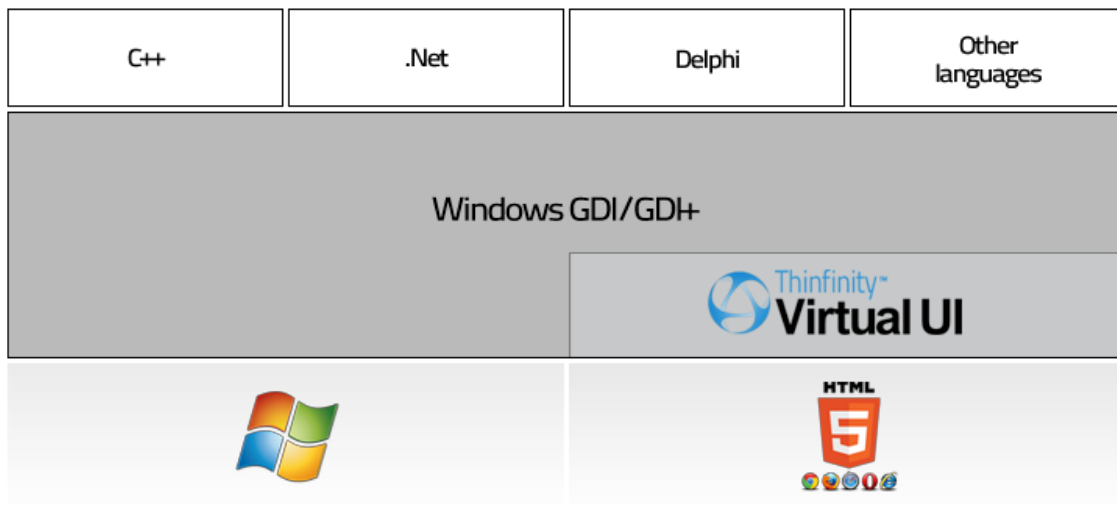
.Net, Delphi and ActiveX

1. Introduction

Remoting Windows Applications is one of the keys to expand the application availability to end-users and boost business efficiency, slashing IT costs and simplifying administration. Thinfinity™ Virtual UI enables applications to be delivered as cross-browser, cross-device web apps, without the need of costly virtualization or remoting environments such as Citrix XenApp® or Microsoft™ RemoteApp.

With Thinfinity™ Virtual UI you can:

- Instantly create dual-platform Windows/HTML5 Apps built in .Net, Delphi, Visual C++, Virtual Basic or others.
- Immediately upgrade and modernize legacy Windows applications.
- Expand application availability to reach a wider user base.
- Deliver applications instantaneously as cross-browser, cross-device web apps
- Deliver your software to customers as a managed service on the cloud
- Dramatically reduce the Total cost of ownership (TCO), by slashing IT costs and simplifying administration.



2. Architecture

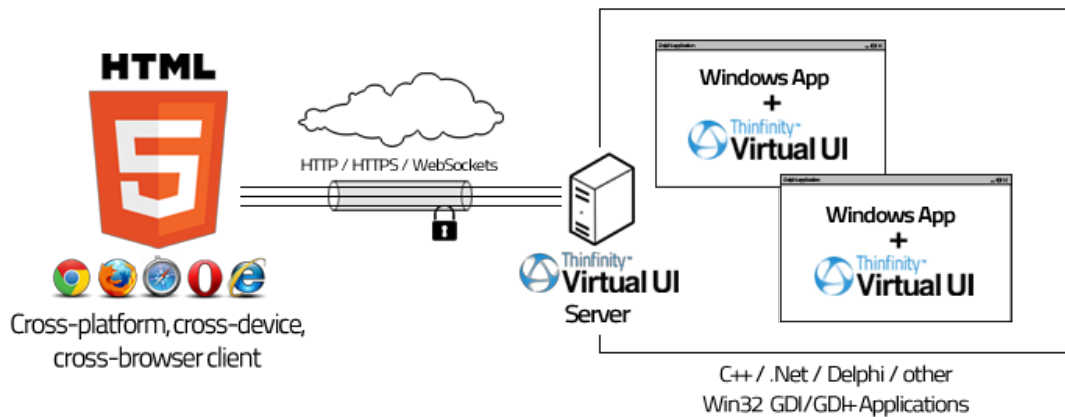
Thinfinity™ Virtual UI is composed of:

- Thinfinity™ Virtual UI SDK
- Thinfinity™ Virtual UI Server
- Thinfinity™ Virtual UI Javascript Client

Thinfinity™ Virtual UI SDK is a set of dlls and/or source code files libraries that plugs into the developer's programming framework to redirect Windows calls and drawing commands to the remote HTML5 canvas.

Thinfinity™ Virtual UI Server is a http(s)/websockets server that communicates with the Windows app through the Thinfinity™ Virtual UI SDK libraries, taking the GDI/GDI+ redirection commands to the web browser.

Thinfinity™ Virtual UI Javascript Client is the responsible for the actual drawing on the web-browser canvas and interacting with the end-user mouse and keyboard events.



Enabling Thinfinity™ Virtual UI in your project is as simple as adding one line to your source code:

C# Example

```
using System;
using System.Windows.Forms;

namespace MyApp
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Cybele.Thinfinity.VirtualUI.Instance.Start();

            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Delphi Example

```
program MyApp;

uses
  Windows,
  Forms,
  Thinfinity.VirtualUI.AutoRun,
  MyApp.Main in MyApp.Main.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

3. Execution behavior

The application execution behavior will depend on how the application is run.

If you run the application from:

- **Window Shell**
When the application is executed from the Windows Shell, it will behave as a standard Windows application.
- **Development Environment**
When the application is executed under a Development Environment (such as Microsoft Visual Studio or Embarcadero's Delphi), an instance of Thinfinity™ Virtual UI Server will be started running in development mode, and the application will be seen both, as a standard Windows application and as a Web application.
- **Web Page**
If the application is launched from a Thinfinity™ Virtual UI Server's page, it will run as a web application.

4. Requirements

These are the requirements for each component of the Thinfinity™ Virtual UI architecture:

Development machine

- Operating Systems:
 - Microsoft Windows 7 32bit / 64bit
 - Microsoft Windows 8 32bit / 64bit
 - Microsoft Windows Server 2012 32bit / 64bit
- A development environment
 - Microsoft Visual Studio
 - Delphi 5 to XE6
 - Microsoft Visual Basic
 - PowerBuilder
 - other
- The application project must use GDI or GDI+ calls. .NET WPF projects are not supported yet.

Server machine

- Microsoft Windows 8 32bit / 64bit
- Microsoft Windows Server 2012 32bit / 64bit

End-user machine

- Any operating system and/or device with a HTML5-compliant Web Browser
- Any modern Web Browser (HTML5-compliant) such as IE10/11, Chrome, Safari, Firefox, Opera, etc.

5. Conclusion

Thinfinity™ Virtual UI allows developers to:

- Instantly upgrade and modernize legacy applications.
- Create dual-platform Windows/HTML5 applications effortlessly, by adding only one line of code to their existing projects built in .Net (WinForms), Delphi, Visual C++ and others.
- Expand applications' availability by delivering them normally on a Windows environment, or by installing them on a Thinfinity™ Virtual UI Server environment to be accessed remotely from any HTML5-compliant Web browser.
- Reduce dramatically the total cost of ownership (TCO), by slashing IT costs and simplifying administration, avoiding costly virtualization/remoting solutions.

Complete Reference:

Find the complete reference on the **Thinfinity™ Virtual UI** Guide under the link:

<http://www.cybelesoft.com/helps/thinfinity/virtualui>

Product Page:

<https://www.cybelesoft.com/Thinfinity/VirtualUI/>