

# Screenshot Web Service

This project demonstrates how to build and deploy a web service to a remote machine. If you have already installed the RTE with web service and web server components on the target, then you can build the non-RTE version of this installer to deploy. Otherwise, use the 'with RTE' version for the initial deployment.

This project provides several web services in one package.

## 1. Remote Screenshots

This function allows you to take a screenshot of the target machine and stream it to any web browser. This service supports jpeg, png and bmp file formats. NOTE: the target machine must have an active user logging in and rendering a screen. If the screensaver is running or they are not logged in via RDP, then there is nothing to grab and therefore you will get a 'screen not rendered' message. This uses .NET calls to get the screen image. If you know a better way to do this, PLEASE let me know!

To call this web service use the following format:

`http://hostname/screenshot/get_image/{png|jpeg|jpg|bmp}`

ex:

`http://192.168.1.1/screenshot/get_image/jpeg`

## 2. Test the service

This function simply echoes back what you send it to test that the service is up and running. The data is returned as an XML structure. To call this service use the format:

`http://hostname/screenshot/echo/{your text here}`

ex:

`http://192.168.1.1/screenshot/echo/hello world`

## 3. Get image of LV panel

This function will get the image of any VI currently running on the target machine. The VI must be running under the LV dev environment or a deployed EXE AND the VI Server must be enabled and set to the default LV port (3363). You also must know the full VI name (not the title, but the actual name in memory that you would use to open a reference to the VI). If the VI is not in memory, you will get an error message. NOTE: you also will specify the image format. To call this service use the format:

`http://hostname/screenshot/get_panel/{png|jpeg|jpg|bmp}/{full VI name}`

ex:

[http://192.168.1.1/screenshot/get\\_panel/png/my app.vi](http://192.168.1.1/screenshot/get_panel/png/my app.vi)

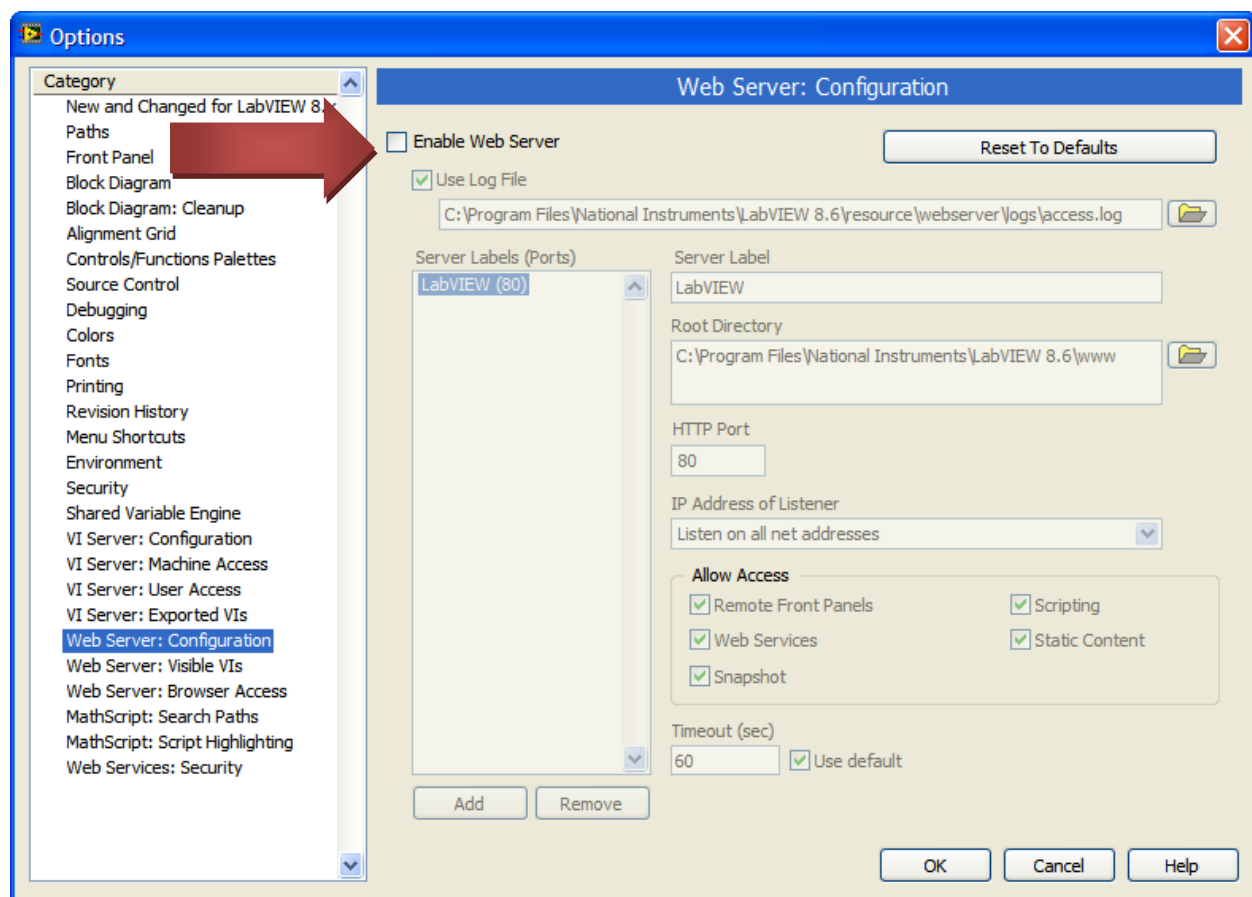
Additional notes:

hostname is the IP address of the target machine where the web service is deployed.

You must open ports 80 and 3363 on your firewall software.

\*\*\*\*\***WARNING**\*\*\*\*\*

**WHEN BUILDING THE WEB SERVICE, EXE and INSTALLER, YOU MUST DISABLE the LV Web Server on the OPTIONS SCREEN!!! Otherwise the niwebserver.conf file will be overwritten by your DEV env settings and it WILL NOT WORK when deployed.**



\*\*\*\*\***WARNING**\*\*\*\*\*

There is a special line in the niwebserver.conf file that is required to make the MIME streaming of jpegs and pngs work. It is:

**LimitResponseBody 10240000**

This is normally set and is an issue with the LV web server. Adding this line works around the issue. NI Support provided me this line to fix the bug I was seeing.

When installing this service, a small exe will be placed in the Startup folder. You must run this EXE and leave it running for the web service to work. I did not bother making this a windows service but if you want to do that, I would be interested in seeing how it is done.

The information used to build this project was acquired from the following web page:

<http://zone.ni.com/devzone/cda/tut/p/id/7747>

One final note: If you redeploy this web service several times after making changes, you will build up a series of deployments on the target that will not be cleaned up. I have noticed that sometimes the wrong version will start after installation. The way to work around this issue is to clear out the old deployments before starting the new one. You can find them on the target machine at:

C:\Documents and Settings\All Users\Application Data\National Instruments\Web Services\UserServices\deployed

I hope you find this project useful as both an example of deploying web services and as a useful service for remote monitoring your deployed apps.

-John Lokanis

PS: Bonus code: 'call web service.vi' shows how to call a LV web service from LV. This is something that NI says they don't know how to do yet. This VI uses some simple .NET calls and the LV XML parser to accomplish this.

\*Unfortunately all of this web service stuff only works on Windows machines at this time.