

NI Web Service Server and the 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 logged 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. Also, since the server runs as a windows service, it must call a VI running inside an EXE that the current user started. This is because the code uses .NET calls to get the screen image. If you know a better way to do this, PLEASE let me know! So, to make this work, be sure to place the 'get local screenshot.vi' in your EXE that is run by the user. I recommend placing it on a part of the block diagram that does not get executed. This should keep it in memory so the web service call can use it. This web service has an optional port parameter that lets you specify the VI Server port your 'user launched' EXE is listening on. If you omit this, the default is 3363.

To call this web service, use the following format:

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

ex:

`http://192.168.1.1/screenshot/get_image/jpeg/3363`

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 port you pass in. If you omit the port parameter then the default LV port (3363) is assumed. 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}/{port}`

ex:

`http://192.168.1.1/screenshot/get_panel/png/my app.vi/3363`

4. Get EXE Version

This function will get the version of any LV EXE currently running on the target machine. You must pass in the VI Server port number if it is not 3363 (the default).

`http://hostname/screenshot/version/{port}`

ex:

`http://192.168.1.1/screenshot/version/3364`

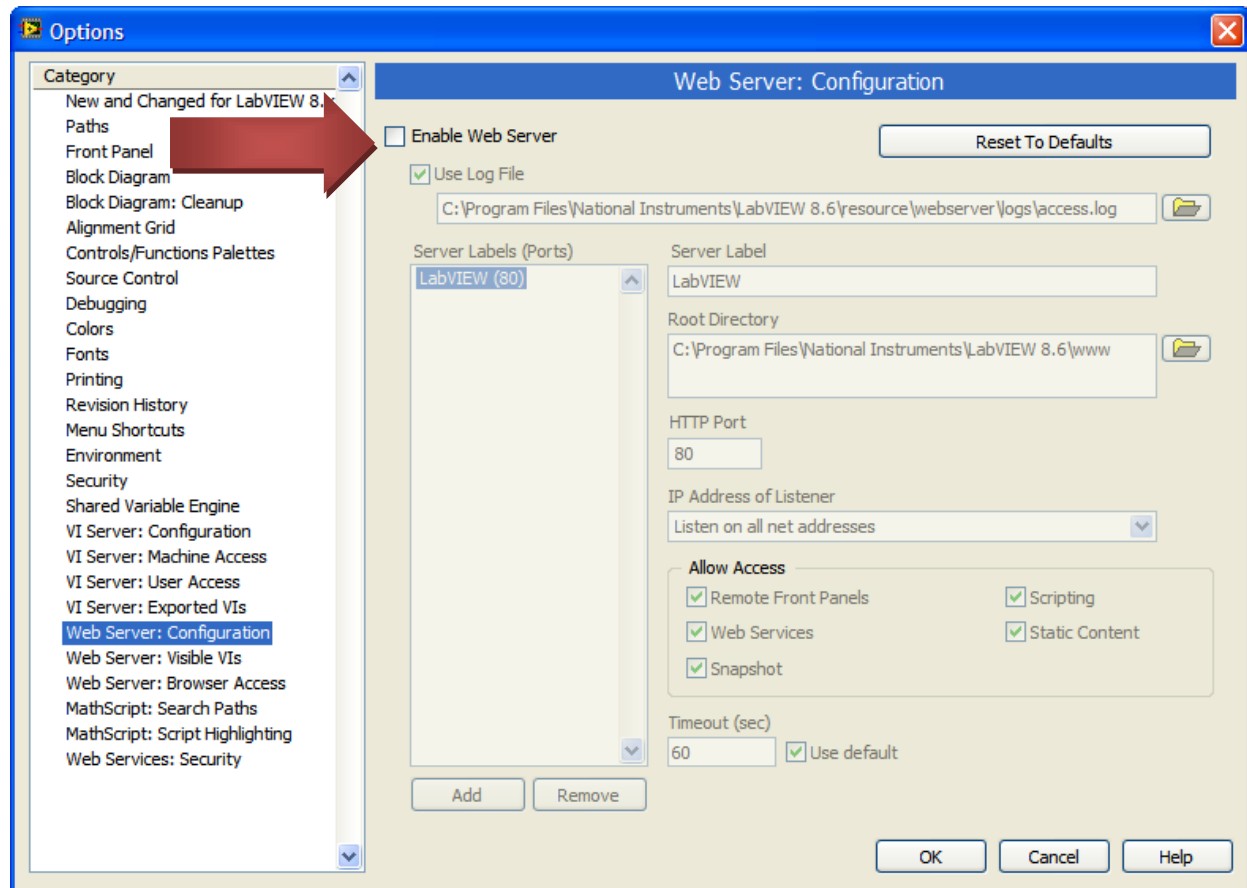
Additional notes:

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

You must open ports 80, 3363 and 3364 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 batch file will run at the end of the installation. This in turn calls a few additional batch files and adds a key to the registry. Here is a summary of what it does:

1. Create the missing logs folder on the target. (Addresses a bug in the web service RTE).

2. Starts the Web Server Service if it is not already running. This ensures that the recently installed LVWS file gets deployed.
3. Stops the Web Server Service.
4. Delete any old installs of the screenshot web service from the target machine. The most recently deployed version is not deleted.
5. Copy some required files to c:\reskit to support running the exe as a service.
6. Merge into the registry the required key for this service.
7. Starts the Web Server Service.

If you want to learn more, check out this link:

<http://zone.ni.com/devzone/cda/tut/p/id/3185>

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

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

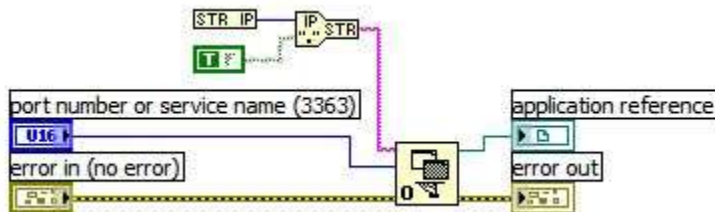
Once you have installed this service on your target, you can add additional web services to the target with other LV installers. You only need to place the LVWS file for your new web service in the "C:\Documents and Settings\All Users\Application Data\National Instruments\Web Services\UserServices\install" folder and the web server will automatically deploy them if it is running. Do not enable the web server in any of your other EXE ini files, however.

One final note: If you redeploy a 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 be running 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

To clean these up, run the wscleanup.bat after installing your other web service installers. Pass it the name of the web service as a parameter.

If you design a web service that you want to call and pass data to an EXE, be sure to wire the actual IP address of the machine to the open application instance. Also, the EXE must have its VI Server running on a different port than the NI Web Service Server (it is using port 3364). Here is an example of how to open the app instance:



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.