

This is the distribution kit for the stand-alone WriteLog internet server.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Building and/or Installing the Server:

The server sources and a Win32 build of each are maintained on github:

<https://github.com/w5xd/WriteLogInetServer>

It has these directories:

- \bin
Contains ContestQsos.wsdl -- needed at run time for WriteLog version 10 clients.
- \bin\win32
Contains the win32 executable built with Visual Studio 2013
- \bin\linux32
- \bin\linux64
use the Makefile to build from source
- \source
Contains the C++ source files for the binaries. You may rebuild the server for platforms not included in this kit. cd to this directory, and type "make". The result is a WritelogInetServer executable in the source directory that you must manually copy to your deployment directory.

Enhanced security at the server

This is an advanced topic. Connecting a program on your computer to the internet exposes your machine to increased chances of being compromised by outside hackers. Firewalls must be turned off, at least selectively for this program, and this program reads files off your computer and copies them to callers.

For this reason, the server has the command switch “-bindAll” If you use that switch, then the WriteLogInetServer exposes its services to all incoming connections from all IP addresses and you are taking your chances. Without the switch, the only connections it accepts are from the same machine it is running on.

The WriteLogInetServer *will answer http GET requests for any URL* for its process working directory and its subdirectories. The only string it rejects is “.” in its URLs, otherwise any file in its directories it will serve out over the internet.

The WriteLogInetServer has essentially no tools for security. If you don't want your server hacked, then you are going to have to learn something about internet security that this document is not going to teach.

There are multiple ways to secure this server. One is to find someone that knows how to run the [apache2 web server](#) (or learn how) and run the WriteLogInetServer under apache2's "reverse proxy" configuration. To get started, you must turn on both the apache2 proxy module and the proxy_http module:

```
a2emod proxy_http
a2enmod proxy_http
```

And you'll need to modify the apache2 file proxy.conf with something like this:

```
<Location /writelog/>
    Proxypass "http://localhost:8001/"
</Location>
```

If, in addition, all your WriteLog installs were done with .NET 3.5 or higher, you can configure apache2 to turn https. (All WriteLog 10 installations cannot use https. WriteLog 11 installed on older XP machines do the internet logbook using the old Microsoft SOAP toolkit, SoapToolkit20.exe, which **cannot** connect to https. WriteLog 12 always can use https.) For either http or https, you can let apache2 do the password authentication, using, for example, AuthType Basic and run the WriteLogInetServer without its -u switch and let apache2 check the username/password combination. You will have to know how to set up apache2 to require them, of course.

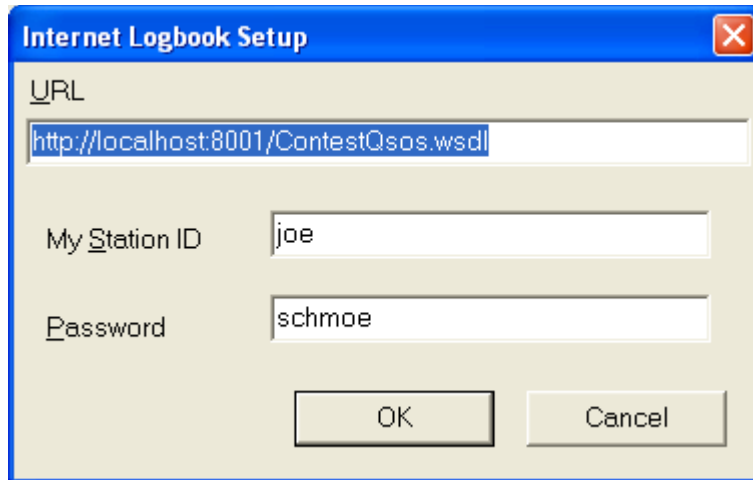
There is one more caveat with https if you're trying to run your server on the cheap with a self-signed certificate. Only WriteLog 11.37 and 12.06 and higher have the option to disable https certificate validation using an INI file entry (see the WriteLog help file for [ScoreUDP] for details. If you run apache2 https with a self-signed certificate (normally you have to pay a signing authority for an https certificate) then only those newer versions of WriteLog have the ability to disable certificate checking.

One more note. If you use Proxypass with the above <Location> directive, then you also have to tell your WriteLog users to connect to that same location in their "Internet Logbook Setup" URL setting, like this: `https://url-of-your-webserver/writelog/ContestQsos.wsdl`.

Another way to secure this server is to use [SSH tunneling](#) with a product like [PUTTY](#).

From the client side

In WriteLog's "Tools Connect To Internet Logbook" menu dialog, there are 3 things you (the server administrator) must tell users to fill in. In that dialog, those three things are:



URL

The url must start with <http://> The next thing in the IP address. It can be numeric, as in 192.168.100.1 or it can be a name, like:

myserver.mydomain.com

You have to know things about DNS to make this work. Note that services like this: <http://www.no-ip.com> can rent you a domain forwarding service, and they even have a free service with not many features, but enables publishing a logbook server from your home PC--if your home network AND your ISP are all configured to allow such publishing.

The domain name must be followed by a colon character and then a port number. The port number must be a number from 1 to 65535, but there are reasons to use various port numbers that are too detailed to document here. If you have no clue, try 8001, the default.

The port number must be followed by a slash character. For WriteLog version 10 clients, the slash character must be followed by the name of a .wsdl file that is located on your server. The one provided is named ContestQsos.wsdl. Here is an example:

`http://192.168.1.100:8001/ContestQsos.wsdl`

My Station ID

The string the user entered here is used in two different places by the system:

- a) all QSOs logged have this string in the "INET" column.
- b) it becomes a "username" to be looked up in your server's password file (see below). If you have no password file (and none is provided in the installation kit), then any username and password is allowed.

Password

If you have a password file at your server, the user must provide the password that matches the username (My Station ID), in your password file. If you have no password file, then the Password entry by users is ignored.

Note that this server uses http basic authentication to send the username/password, which means they are not encrypted as transmitted over the internet. This will only keep "honest" people out of your server. Anyone with the ability to sniff your tcp packets between your server and your clients can read the usernames/passwords. Make up dedicated usernames and passwords for this application that you use nowhere else.

Configuring the server.

It is designed to run with the process current directory set to the \bin installation directory. ContestQsos.wsdl (an edited and renamed version of it) must be in the process current directory. (ContestQsos.wsdl is only used by WriteLog clients that used the SoapToolkit20.exe installation. Newer versions don't need this wsdl file.)

Editing the .wsdl file

This server does not use the .wsdl, but some clients require it at run time, and it is served from WritelogInetServer. ContestQsos.wsdl **must** be edited for your installation. At the bottom, you will find this line:

```
<soap:address location="http://192.168.1.101:8001"/>
```

You must change the address (the 192.168.1.101) to the IP address your clients are to connect to. You *may* (and usually *should*) use the TCP name rather than the numeric address if your server is located where DNS will have the right number. The number after the colon, :8001 in this example, must match the "-p 8001" used to launch this program. Note that WriteLog users must also enter the file name in their setup. That means the server *may* have multiple .wsdl files if you have the situation where some users route through, say, a NAT server and others are "inside" the NAT. Have two different WSDL files and tell each user which one to connect to.

4. A username/password file.

This file is optional. There is none provided; you have to create one with a text editor. If you create one, launch the executable with the -u parameter pointing to it. You may place it anywhere in the directory structure that you want, but be aware that if you put it in the \bin installation folder or any subfolder, the server will allow it to be downloaded by any internet browser. Not having a password file means no usernames or passwords are required to connect to the server.

More than one "user" is allowed to connect simultaneously using the same username and password combination, but, generally, you want to assign different ones to every connecting station because the username at the WriteLog end is logged with each QSO.

5. SHUTDOWN file

If there happens to be a file in the process current directory who's name contains the string SHUTDOWN, and if a user with a valid username/password uses a web browser to brose to that file, then this server will server out the contents of that file and then shut down. Note that files of any other name in this directory or any subdirectory are available to browsers.

Running the server

usage: WritelogInetServer [-h] [-u <userFile>] [-p <portNumber>] [-v]

-v
turns on the verbose flag that prints sent/received QSO counts.

-h
prints this help message

-u <userFile>
<userFile> is the path to a username/password file.
The file is formatted with one username/password pair per line.
Separate the username from the password with one space character.
These become the legal entries from WriteLog users in their
Connect to Internet Logbook dialog.

-p <portNumber>
This program requires a TCP port. The default without -p is 8001.
The number must be in the range of 1 through 65534. Different operating
systems, internet providers, and routers will allow or not allow various
port numbers through.

-t <number of threads>
number of threads to use to serve. Limited to 10 max. Default is 1.

-L6
turns on the limit-labels-to-6-characters flag, which then rejects invalid
input
from WriteLog versions earlier than 10.76.

-bindAll
enables internet connections on all IP addresses for this machine.

If WriteLog version 10.75 and earlier are participating in the network, then you should also add the "-L6" command option. This option causes the server to truncate network frequency names longer than 6 characters, which those earlier WriteLog versions can send to the server.

That's all. If you add -v to the command line, it prints messages as it goes along. If it exits, you can simply restart it. The WriteLog clients will re-populate the logbook after it restarts. For unattended use, write a script that simply runs WritelogInetServer in a loop-

if it crashes (which it should not), the script just starts it over. Such scripts are operating-system specific and not provided in this kit.

Test the server from a browser

A web browser can be used to test whether the server is running and can answer requests on its selected ports. Browse to the same URL that WriteLog users are instructed to connect to in WriteLog:

`http:// myserver.mydomain.com:8001/ContestQsos.wsdl`

The result should be a view of the ContestQsos.wsdl file as edited per these instructions. If you have configured the server to require passwords, your browser will ask for one of them before it will allow a download of the file.

History

May, 2016

update CPP server to use synchronization and build using C++11 for both Linux and Win32.

WritelogInetServerKit06.zip December, 2011
Support WriteLog version 11 clients.

WritelogInetServerKit05.zip May, 2011
update documentation
use timeouts to avoid hung server
verbose option prints incoming IP address

WritelogInetServerKit04.zip February, 2010
add -L6 option

WritelogInetServerKit03.zip February 2010
fix 5 minute delay overwriting old network frequency with incoming.