BUILD NOTES for Foobar2000 foo_controlserver plug-in

August 2017 - Walter Hartman

Foo_controlserver is a Foobar2000 plug-in enabling remote control of Foobar2000 via an Ethernet socket connection.

You would use it in a setup composed of:

Foobar2000 running on a server PC on your local network, and with the foo_controlserver plug-in installed and configured.

You would then use a remote client app to connect to and control Foobar2000 over your local network.

You can use any remote client app that sends and receives commands defined for foo_controlserver. See below for a list of commands and the source code for specifics.

One such remote client app is "Foobar2000 Copilot". This is a mobile app I wrote for Microsoft Windows Phone, version 8.0 or above.

For more details, see : http://www.foobar2000copilot.com

---

foo_controlserver version 1.0.2 – foobar2000 plug-in to control foobar2000 over TCP/IP

Version 1.1.3 November 2016 - Walter Hartman

This version of the software is released under the same terms as the original GPL license -- see above. As per the original terms, this software is free and full source code for this version is provided, along with the Foobar2000 plug-in binary 'foo_controlserver.zip' – thus keeping it in the public domain.

Changes in version 1.1.3 include the following:

- Rebuilt using VS2015 Community

- Rebuilt using new foobar2000 SDK version 2015-08-03

- Updated the preferences dialog to use the new foobar2000 SDK APIs. You will no longer see the 'this is a legacy preferences page' warning.

- Added IP address display of local PC to preferences dialog

- Added get album art command, returns image (jpg or png), encoded as a base64 string

- Added media library search command

- Fixed some bugs related to the 'now playing' track

To do the build the source code, you will need the following software (all free) :

1- Visual Studio 2015 Community

2- Foobar2000 SDK version 2015-08-03

3- Microsoft Foundation Class - MFC

4- WTL9.1 - Windows Template Library

This version of foo_controlserver component was built in the free version of Microsoft Visual Studio Community 2015 in C++.

1) In VS2015, use nuget to install the WTL (Windows Template Library) -- needed for the Foobar2000 ATLHelpers library and to build the new preferences screen.

From VS2015, select **Tools>NuGet Package Manager>Manager Nuget Packages for Solutions**, then from that dialog, select **'Browse'** and search for **WTL**.

This build was built using **WTL version 9.1.0**

2) Also make sure the MFC libraries were installed in your VS2015 Community setup.

That option is available when doing a 'custom' install under :

Programming languages> Visual C++ language > Microsoft Foundation Classes for C++

If the MFC classes were not installed in your initial VS2015 setup, you can modify your Visual Studio feature set afterwards by going to :

Control panel>Programs>Programs and Features>Microsoft Visual Studio Community 2015>Change

3) Download the Foobar2000 SDK from http://www.foobar2000.org/SDK

Version SDK 2015-08-03 was used for this build.

Unzip that SDK into a work directory. Place the 'foo_controlserver' folder for this project into that work directory at the same level as the Foobar2000 SDK, shared, ATLHelpers directories as follows:

```
 ./SDK-2015-08-03/foobar2000/
                  ./SDK
                  ./shared
                  ./ATLHelpers
                  ./helpers
                  ./foo_sample
                   ./foo_controlserver   <---
```

Make sure you have the Foobar2000 SDK libraries built before attempting to build foo_controlserver since it depends on the SDK libraries.

Relative directory paths for various includes used in foo_controlserver depend on the above directory structure.

4) ./foo_controlserver/src/foo_controlserver.sln is the VS2015 solution file for this project. Open that file to start. Make sure you install the WTL9.1 package as described above since the .sln references it.

Build a Release version of the code in VS2015 -- that will generate the foo_controlserver.dll in the **Release** directory. Then in Windows, right click on that file, and **select Send To > Compressed (zipped) folder**.

This will create **foo_controlserver.zip**. This is your install file for the Foobar2000 component -- this is the only file that the end user needs to install the plug-in.

Foo_controlserver commands (see source code, for more details):

next [playlist#]

- play next track in the playlist

prev [playlist#]

- play previous track in the playlist

play [track#] | [<playlist#> <track#>]

- play current or item # from the playlist

rand [playlist#]

- play a random track from the playlist

seek ['delta'] <secs>

- seek to specified time or by delta time

pause

- pause/unpause current track

stop

- stop current track

list [playlist#] [<track#> <track#>]

- list playlist tracks between range

libsearch 'playlist name' 'string'

- media search for info, load results into playlist 'playlist name', use quotes on strings – playlist will be created if it doesn’t already exist

listinfo [playlist#] | [<playlist#> <playlist#>]|['all']

- info about current playlist or all lists

serverinfo

- info about the server

trackinfo

- info about the current track

vol [#] | ['up'] | ['down'] | ['mute']

- get/set volume dB or up/down 0.50 dB or mute

order [type]

- get/set order type [default|random|repeatplaylist|repeattrack|shuffletrack|shufflealbum|shufflefolder]

queue [[playlist#] <track#>]

- queue track from playlist or view queue

queue del <index#> | <'all'>

- delete queue index item from queue or clear all queue

albumart

- get album art for now playing track, sends back jpg or png image as a base64 encoded string