

Harmonic (Rhozet) Carbon FFMPEG Exporter PlugIn

About

This product is intended but not limited to fill performance or format gaps of the “Out of the box” Carbon Coder Application.

Whilst the development phase of this plugin, we recognized, that there are too much special features that are very dependent on usecase, that it is most likely that if you encounter missing features or problems with this plugin with special formats, we will introduce a spin-off of this plugin to support special features of a specific codec combination as a new product. Anyway, these sentences shall not discourage users to submit bug reports and/or feature requests.

Legal information

Please make sure the “shared” builds of ffmpeg and its prerequisites you download or build yourself are licensed properly.

The libraries that are dynamically linked by this dll (RPI Plugin) are licensed within the GPLv2 License. Within this license, it is explicitly allowed to “dynamically link” to such libraries from proprietary software. As we do not make use of any code that comes with ffmpeg or any of its libraries within our dll, it does not underly the GPLv2 or any other open-source License. Users of this RPI plugin are allowed to reverse engineer or debug all deliverables of ffmpeg whilst our plugin makes use of them. Also we will not spare with answers on questions about what individual methods we make use of, if this is of interest for you.

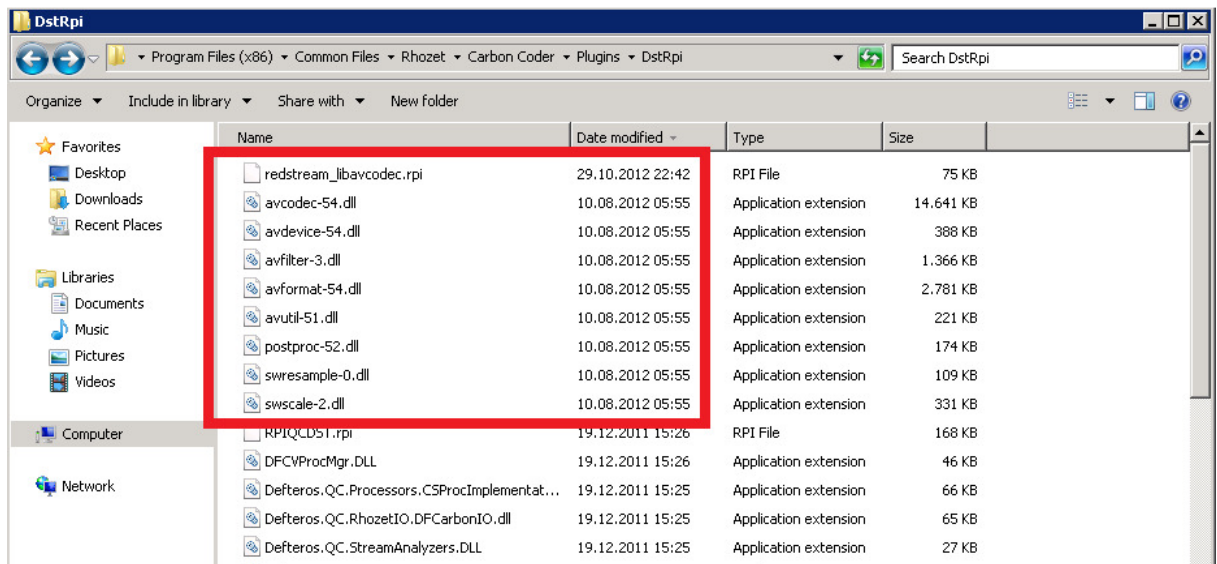
Must Know - Restrictions

The Plugin was not yet optimized to be used from the local Carbon Coder application. When playing around with output formats and settings, please resart the application on every major change like container or encoder setting. Otherwise you will encounter freezed carbon application, an output file that does not match your output settings or worse. Use “project - pcp” save and load to speed up your work.

Installation – General

Do this on each individual node before trying to make use of the Plugin from WFS

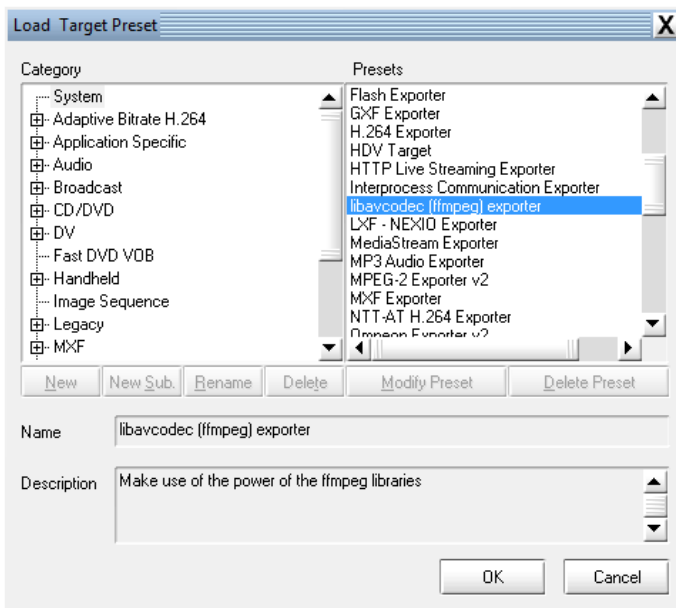
- This Version is tested against a standard shared build from <http://ffmpeg.zeranoe.com/builds/win32/shared/ffmpeg-20120810-git-633b90c-win32-shared.7z>
 - ffmpeg-20120810-git-633b90c-win32-shared
- You may try or use any other version, but it is highly recommended to start with this one.
- The Plugin itself and all the dlls from ffmpeg go to the DstRpi Directory of each Carbon Node where you want to make use of it, on a Server 2008 this is:
 - C:\Program Files (x86)\Common Files\Rhozet\Carbon Coder\Plugins\DstRpi



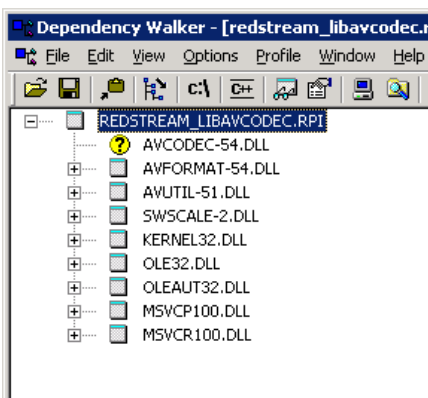
- After copying the libraries into this folder, restart the “Nexus Service” within the Windows Services, also close all active CarbonCoder Applications like PNXKernel and PNXServer.exe, Carboncoder.exe.
- Make sure at least these libraries are in the DstRpi Folder of your Carbon:
 - avcodec-54.dll
 - avformat-54.dll
 - avutil-51.dll
 - swscale-2.dll
 - It is not recommended to rename ffmpeg dll’s from other versions to match these requirements, better download the right ones.

Installation – Debugging

You recognize when the installation is finished if you start the local Carbon Coder Application and see the “libavcodec (ffmpeg) exporter” available in /System . For every installation retry, you have to restart the Nexus Service again:



You can also make use of a free program called “Dependency walker” in order to prove that all needed dependencies are there. Just drag the libavcodec_rpi_exporter.rpi file into the dependency walker window. In the Screenshot below, the AVCODEC-54.DLL is missing in the same directory as the exporter dll resides in, all other dependencies are there. (Kernel32 and others are Microsoft dependencies and there anyway)

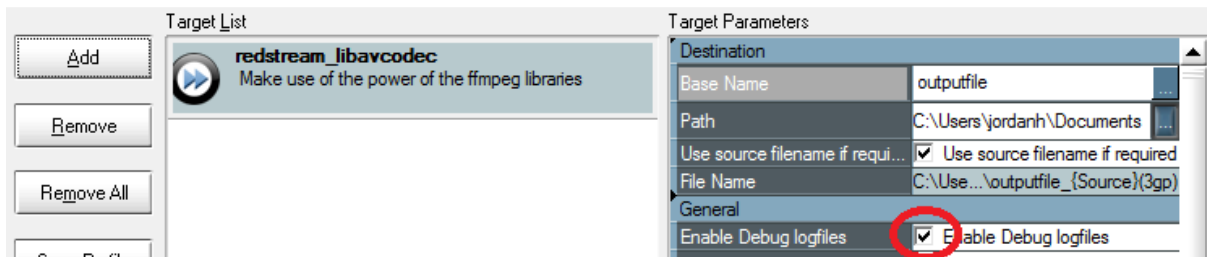


Transcode settings – Debugging

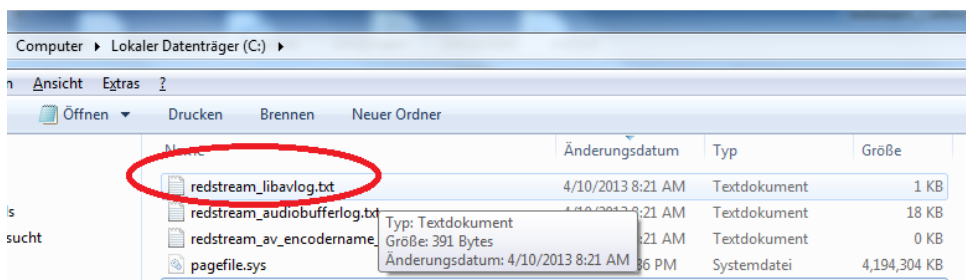
While you play around with output settings using the local Carbon Coder Application, please do never forget, that the PlugIn is not designed for multiple usage in one runtime. Unfortunately, the local Carbon Coder Application does not create a new instance of its PlugIns for each transcode session. You will experience strange behavior when changing some settings and re-using the same runtime again. Please always save your current Carbon Project before hitting the start button. If the application hangs, kill and restart it and try the same settings again.

In case, you are sure to have settings that should work (settings match to the standard of your chosen wrapper/codec combination), you may enable debug logging to see the messages of libavcodec.

Activating the “Enable debug logfiles” checkbox will give you more insight about whats going on



- Debug log files should be written to the c:\ drive:



- The carbon application should be restarted EVERY Time the conversion stopped, otherwise it will most likely hang

The really interesting log for users of this plugin is the libavcodec_libavlog.txt file. It contains all the messages that you would also see when using ffmpeg.exe on commandline. This should at least give you a hint about where the problem is, or at least a direction: “could not open audio codec” for example should mean, that you need to change some audio settings

WFS Deployment

1. In WFS or Carbon Server environment, you always should choose a special node that does not have any other Management functions in order to configure any Userprofile or RPI Plugin.
2. Don't forget to restart the Nexus Service on all Machines where the Plugin dll's are deployed.
3. Go to your WFS Manager and update the Presets (File-> Preset Connections). Like mentioned before, you should always connect to the same Node (your "configuration" node) when doing this.
4. Restart the WFS Manager application after this procedure.

Version

Version 1.0

DEMO Limitation: 1000 Input Frames

Copyright © 2013 x-dream-media GmbH