

PrimeNetEncoder

PrimeNetEncoder is a network encoder for HDHomeRunPrime. It is designed to run on both windows and linux and to have minimal requirements. The encoder uses the hdhomerun_config application and ffmpeg to capture the stream. Since ffmpeg is only capturing the stream and writing to a file PrimeNetEncoder has very low overhead.

PrimeNetEncoder can be configured to handle any number of HDHomeRun Prime tuners. If you are going to use more than one HDHomeRun Prime you will need to modify the tuners.count property, and create new entries for the tunners in the PrimeNetEncoder.properties file, and create new tuner entries in the sage.properties. Forum members have tested and confirmed that PrimeNetEncoder can also work with other HDHomeRun tuners. Please check the forums for notes on compatibility.

PrimeNetEncoder also offers the option to transcode to h.264 in realtime. This does requires a reasonable amount of CPU resources to accomplish. Transcoding is configurable at the tunner level. There are settings available for quality and performance.

Preqs:

1. Java 1.6 or above must be installed and working
2. HDHomeRun Prime must be installed and working with channels being tunable using the HDHomeRun software
3. If you are running firewall software you will need to open the following ports (*Or more depending on the your configuration*)
 - a. Machine running PrimeNetEncoder
 - i. Outbound
 1. TCP/7000-7002 (*Network Encoder*)
 - ii. Inbound
 1. UDP/5000-5002 (*HDHomeRun*)
 2. UDP/8271 (*SageTV Discovery*)
 - b. SageTV Server
 - i. Inbound
 1. TCP/7000-7002 (*Network Encoder*)
 - ii. Outbound
 1. UDP/8271 (*Discovery*)

Known Issues:

1. When using the transcoding feature if there is not enough resources for ffmpeg to keep up with transcoding the stream in realtime it will just stop. You will get a no signal in Sage until sage realizes the stream has stopped and it restarts the stream again. I recommend only using transcoding if you have appropriate resources on your machine.
2. Using the PrimeNetEncoder as a runnable class from SageTV has two drawbacks in version 7 of SageTV. The first issue is that network encoder discovery has already been executed, and SageTV will not be able to discover PrimeNetEncoder. The second issue

is that SageTV may not see the tuners as being available until the users goes to Video Sources and selects the tuner. At this time I do not recommend running PrimeNetEncoder as a runnable class. I am testing changes in V9 of SageTV that fix these issues.

Setup:

1. Download PrimeNetEncoder package and extract the contents.
2. Download appropriate [ffmpeg](#) package for your platform (I am currently using 2.6.1)
 - a. Place ffmpeg.exe in PrimeNetEncoder folder
3. Determine the device ID of the HDHomeRun Prime(s) that are going to be used
 - a. From a command prompt navigate to the HDHomeRun prime software folder. On windows it will usually be C:\Program Files\SiliconDust\HDHomeRun
 - b. Execute HDHomeRun_config.exe discover
 - c. Execute the PrimeNetEncoder so that it can create a default properties file. Then close PrimeNetEncoder.
 - d. Modify the following lines in PrimeNetEncoder.properties file that is created from the step above. Set the tuner(index).id to the id returned from hdhomerun_config.exe. Set the HDHomeRunConfig.path to the full path to the hdhomerun_config.exe

```
tuner1.id=FFFFFFFF
tuner2.id=FFFFFFFF
tuner0.id=FFFFFFFF
HDHomeRunConfig.path=hdhomerun_config.exe
```

4. Update sage properties file with properties to enable network encoders and network encoder discovery.
 - a. Some of these properties may already be appropriately set

```
encoding_discovery_port=8271
network_encoder_discovery=true
```

5. There are two options for installation.
 - a. Option 1 is to run PrimeNetEncoder in console mode. If the machine you want to run PrimeNetEncoder on is not your SageTV server this would be how you want to continue.
 - i. From a command prompt make sure your current directory is the PrimeNetEncoder directory
 - ii. Execute java -jar PrimeNetEncoder.jar
 - b. Option 2 is to run PrimeNetEncoder as a runnable class from the SageTV server.
(Note: This method is not currently recommended. I am working on some changes so this should work properly in version 9)
 - i. Make sure the SageTV service/application is not running

- ii. Copy PrimeNetEncoder.Properties and ffmpeg.exe file to the SageTV directory
- iii. Copy PrimeNetEncoder.jar to the JARs folder in the SageTV home directory
- iv. In the sage.properties file modify property load_at_startup_runnable_classes.
 - 1. Add jvl.primenetencoder.PrimeNetEncoder to the list
- v. Start SageTV service application.
 - 1. NOTE: The runnable classes is one of the last items that Sage starts. Once the clients are available it should be started.
- c. From the sage TV interface add the tuners.
 - i. Configure each tuner and select a guide.
 - 1. Make sure there is no logical channel mappings or SageTV will send them instead of the virtual channel number.
 - 2. Note: Previewing channels when adding the tuner will not work with the Network Tuner. Just skip the preview step.

PrimeNetEncoder Console Commands:

status – Prints a list of each of the tuners and there current status

status [tuner index] – Prints details about the current tuner

echologs – Echoes all of the information that is being written to the logs file to the console.

Press return to stop echoing.

Realtime Transcoding

The properties below control the realtime transcoding to h.264. To enable transcoding set the enabled property to True. The bitrate property controls the file size and quality of the recording. The preset controls how efficiently the encoder uses the space. Look at the documentation of ffmpeg for further information on these settings.

```
tuner0.transcode.enabled=False
tuner0.transcode.deinterlace=True
tuner0.transcode.bitrate=4000
tuner0.transcode.codec=libx264
tuner8.transcode.scaling=-1:720
tuner0.transcode.preset=ultrafast
```

Settings

```
ffmpeg.analyzeduration=2000000
```

This setting controls how long ffmpeg analyzes the stream to determine its contents. I have been fairly successful with a setting of 2000000 on Comcast. ffmpeg's default setting is 5000000.

`ffmpeg.probesize=5000000`

This setting controls the size of data that ffmpeg analyzes to determine the stream info. ffmpeg's default setting is 5000000.