

Capture Sermon

Capture Sermon is a sermon audio recording package that has been developed and in use for <http://audio.priorlakebaptist.org> . It is composed of:

1. a base set of Java programs that take care of the:
 - audio recording and track/sermon marking
 - audio port settings
 - wave file splitting into tracks as well as separate sermon wav file
Tracks are based on user input during recording as well as auto-ensuring that no track is longer than 5 minutes or smaller than 10 seconds.
There's also an option to limit track splitting to the last portion of recording for purposes of fitting the (CD) media record limit
 - cdburner "cdrdao" Table of Content generation
2. plus python scripts that automate many of the higher level actions such as
 - Calling the Java programs in the correct sequence
 - GUI request for Sermon Info such as Title and Speaker
 - Generating Time Stamped file names
 - Controlling the cd burning, the sermon compression
 - Auto-ftping the sermon up to the web site
 - Archiving, and directory clean up

Downloading the software:

The CaptureSermon files are provided "AS IS" [and without warranty, express or implied.](#)

<http://www.priorlakebaptist.org/CS/CaptureSermon.zip>
<http://www.priorlakebaptist.org/CS/CaptureSermon.pdf>

Prerequisites:

- Java Runtime Environment (JRE) 6u1
<http://java.sun.com/javase/downloads/index.jsp>
- (These will be auto-installed by CaptureSermon.py if sufficient file access rights)
http://tritonius.org/tritonius_share-0.3.6.jar
http://tritonius.org/tritonius_remaining-0.3.6.jar placed in JRE's lib/ext directory
- Python 2.3 or newer. Suggest <http://www.activestate.com/products/activepython>
- ([CaptureSermon comes with pre-installed](#) <http://www.voidspace.org.uk/python/configobj.html>)
- <http://www.wxpython.org/download.php#binaries>
- An mp3 encoder such as <http://lame.sourceforge.net/links.php>

- A cd burner such as <http://cdrdao.sourceforge.net>
- A VUM such as windows meterV.exe or gnome/linux vumeter

Installation:

The default installation is to extract to the CSRec directory under the root directory:

linux: cd /; unzip CaptureSermon.zip

windows: extract all to C:\

Windows users can create a desktop shortcut to C:\CSRec\bat which contains the following

CS.bat	Runs CaptureSermon software with timestamped filename
CS_getspeaker.bat	Tests the wxpython dependent getspeaker info GUI
CS_pause.bat	Runs CaptureSermon software with response pauses
CS_redo.bat	Re-runs CaptureSermon software selectively
CS_test.bat	Runs CaptureSermon software using filename TEST

Unix/Linux users can create a desktop shortcut to /CSRec/shell

Setup:

In the bin subdir of CSRec, copy either CaptureSermon.ini_windows or CaptureSermon.ini_linux to CaptureSermon.in and then edit the following entries:
[ftp]

```
site = 'YourWebSiteAddress'
userid = "FtpUserid"
passwd = "FtpPasswd"
audio_path = "/audio"      # ftp audio directory path
audio_prefix="CS"         # audio file prefix
```

[speakers]

```
nlist= speaker 1, speaker2  # (single line) speaker list for CS_getspeaker.pyc
ndefault                    # default speaker
school                      # sunday school start time
```

Make certain all the [tools] and [paths] entries are correct. In [sound] set the mixer according to the mixer name from running set_ports.jar (double click or java -jar set_ports.jar).

Running:

linux: cd /CSRec/bin; python CaptureSermon.py

windows: click into c:\CSRec\bat shortcut and then double click on desired bat file

Selecting Audio Equipment: TBD

Setting Audio Card input line volume: TBD

Appendix: The base Java programs

The CaptureSermon files are provided "AS IS" and without warranty, express or implied.

```
# List various ports
java -jar set_ports.jar
```

```
# options: use -p for setting recording volumes
java -jar set_ports.jar -h
```

```
Usage: set_ports -cp . [options]
where the options are:
-h      : help
-p linename:mute:pan:volume
where mute is T or F or -
pan is number or -
volume is number or -
- => don't change
```

```
java -jar CaptureSermon.jar -h
```

```
Usage: AudioRecorder -jar audiorecorder.jar [options]
where the options are:
-h      : help
-l      : list input capable audio mixers and exit
-m mixer : set mixer
-o filename : output file base name (default test)
```

```
# CaptureSermon.jar will create a filename.wav file
#           as well as a filename.cue file
```

The cue file will look like the following:
more test.cue

```
//sermon [S]tart [E]nd and track [M]arker and [L]ast cue Frame positions
M 518468 <----- frames manually marked as significant track locations
S 1657838 <----- beginning of Sermon
E 3170884 <----- ending of Sermon
L 3596950 <----- Last frame
```

```
=====
#This is necessary to fix bug in JAVA Sounds wave size (at least
JRE1.5 and earlier)
```

```
java -cp . fix_wav -h
```

Usage:

Info: fix_wav <file>

Fix: fix_wav <file> fix

fix_wav [options]

where the options are:

-h : help

Ex:

```
java -cp . fix_wav filename.wav fix
```

```
=====
prerequisite: You need to copy
tritonus_share-0.3.6.jar and tritonus_remaining-0.3.6.jar into jre1.6.0_01/lib/ext/
```

ssplit.jar splits the (last -t secs of the) wave file into tracks based on the *.cue file, further auto-splits any tracks to maximum 5 minute durations, and creates a contiguous sermon file for compression (for the web)

```
java -jar ssplit.jar
```

Usage: java -jar ssplit.jar [options] filename(s) [options]

where the options are:

-h : help

-t secs : tail length

File types supported: WAVE AU AIFF

Ex:

```
java -jar ssplit.jar filename.wav
test.wav Framelength is 3593216
```

```
M 518468
```

```
S 1657838
```

```
E 3170884
```

```
L 3596950 FL: 3593216
```

```
===== processed cue ===== size: 3
```

```
M 518468
```

S 1657838

E 3593216 <----- Sermon swallowed last track as it was less than 10 seconds

splitting into filename_0.wav

splitting into filename_1.wav

splitting into filename_2.wav

=====
It also creates a filename_s.wav (the Sermon)

as well as a cdburner "cdrdao" Table of Contents:

more filename.toc

CD_DA

TRACK AUDIO

COPY

FILE "filename_0.wav" 0

TRACK AUDIO

COPY

FILE "filename_1.wav" 0

TRACK AUDIO

COPY

FILE "filename_2.wav" 0