

OGG/Vorbis

Introduction

The OGG/Vorbis file format, as defined by [» http://www.vorbis.com/](http://www.vorbis.com/), is a scheme for compressing audio streams by multiple factors with a minimum of quality loss. This extension adds Ogg Vorbis support to PHP's [URL Wrappers](#). When used in read mode, compressed OGG/Vorbis data is expanded to raw PCM audio in one of six PCM encoding formats listed below.

Installing/Configuring

Requirements

This extension requires *PHP* $\geq 4.3.0$, [» libogg](#) ≥ 1.0 , and [» libvorbis](#) ≥ 1.0 .

Installation

Information for installing this PECL extension may be found in the manual chapter titled [Installation of PECL extensions](#). Additional information such as new releases, downloads, source files, maintainer information, and a CHANGELOG, can be located here: [» http://pecl.php.net/package/oggvorbis](#)

Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

Resource Types

This extension has no resource types defined.

Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

OGG/Vorbis supports PCM encodings in the following formats

Constant	Definition
OGGVORBIS_PCM_U8	Unsigned 8-bit PCM.
OGGVORBIS_PCM_S8	Signed 8-bit PCM.
OGGVORBIS_PCM_U16_LE	Unsigned 16-bit PCM. Little Endian byte order.
OGGVORBIS_PCM_U16_BE	Unsigned 16-bit PCM. Big Endian byte order.
OGGVORBIS_PCM_S16_LE	Signed 16-bit PCM. Little Endian byte order.
OGGVORBIS_PCM_S16_BE	Signed 16-bit PCM. Big Endian byte order.

Context options

OGG/Vorbis tuning options

Option	Definition	Relevance	Default
pcm_mode	PCM byte encoding used. See constants below.	Read / Write	OGGVORBIS_PCM_S16_LE
rate	PCM Sampling rate. Measured in Hz.	Write only	44100
bitrate	Vorbis Average Bitrate Encoding / Variable Bitrate Encoding. Measured in bps (ABR) or Quality level (VBR: 0.0 to 1.0). 128000 ABR is rough equal to 0.4 VBR.	Write only	128000
channels	Number of PCM channels. 1 == Mono, 2 == Stereo.	Write only	2
serialno	Serial Number of stream within file. Must be unique within file. Because of the potential to select a duplicate serial number within a chained file, make efforts to manually assign unique numbers when encoding.	Write only	<i>Random</i>
comments	Associative array of file comments. Will be translated to <i>strtoupper(\$name) . "=\$value"</i> . Note: This context option is not available in oggvorbis-0.1	Write only	array('ENCODER' => 'PHP/OggVorbis', http://pear.php.net/oggvorbis)

Examples

Examples on using the ogg:// wrapper.

Example #1 - Reading an OGG/Vorbis file

```
<?php
dl("oggvorbis.so");

/* By default, ogg:// will decode to Signed 16-bit Little Endian */
$fp = fopen('ogg://myaudio.ogg', 'r');

/* Collect some information about the file. */
$metadata = stream_get_meta_data($fp);

/* Inspect the first song (usually the only song,
   but OGG/Vorbis files may be chained) */
$songdata = $metadata['wrapper_data'][0];

echo "OGG/Vorbis file encoded by: {$songdata['vendor']}\n.";
echo "  {$songdata['channels']} channels of {$songdata['rate']}Hz sampling
encoded at {$songdata['bitrate_nominal']}bps.\n";
foreach($songdata['comments'] as $comment) {
    echo "  $comment\n";
}

while ($audio_data = fread($fp, 8192)) {
    /* Do something with the PCM audio we're extracting from the OGG.
       Copying to /dev/dsp is a good target on linux systems,
       just remember to setup the device for your sampling mode first. */
}

fclose($fp);

?>
```

Example #2 - Encode an audio file to OGG/Vorbis

```
<?php
dl('oggvorbis.so');

$context = stream_context_create(array('ogg'=>array(
    'pcm_mode' => OGGVORBIS_PCM_S8, /* Signed 8bit audio */
    'rate' => 44100, /* 44kHz CD quality */
    'bitrate' => 0.5, /* Midquality VBR */
    'channels' => 1, /* Mono */
    'serialno' => 12345))); /* Unique within our stream */
```

```
/* Open file for appending. This will "chain" a second OGG stream at the
end of the first. */
$ogg = fopen('ogg://mysong.ogg', 'a', false, $context);

$pcm = fopen('mysample.pcm', 'r');

/* Compress the raw PCM audio from mysample.pcm into mysong.ogg */
stream_copy_to_stream($pcm, $ogg);

fclose($pcm);
fclose($ogg);
?>
```