BASS_OPUS_StreamCreateFile

Creates a sample stream from an Opus file.

```
HSTREAM BASS_OPUS_StreamCreateFile(
    BOOL mem,
    void *file,
    QWORD offset,
    QWORD length,
    DWORD flags
);
```

Parameters

mem	TRUE = stream the file from memory.
file	Filename (mem = FALSE) or a memory location (mem = TRUE).
offset	File offset to begin streaming from (only used if mem = FALSE).
length	Data length... 0 = use all data up to the end of the file (if mem = FALSE)
flags	A combination of these flags.

BASS_SAMPLE_FLOAT	Use 32-bit floating-point sample data. Floating-point channels for info. If thi flag is not specified, then the stream is bit.
BASS_SAMPLE_SOFTWARE	Force the stream to not use hardware mixing.
BASS_SAMPLE_3D	Enable 3D functionality. This requires that the BASS_DEVICE_3D flag was specified when calling BASS_Init, an the stream must be mono. The SPEAK flags can not be used together with thi flag.
BASS_SAMPLE_LOOP	Loop the file. This flag can be toggled any time using BASS_ChannelFlags.
BASS_SAMPLE_FX	Enable the old implementation of Dire 8 effects. See the DX8 effect implementations section for details. U BASS_ChannelSetFX to add effects t stream.
BASS_STREAM_PRESCAN	Pre-scan the file for seek points and accurate length reading in chained Op files (has no effect on normal Opus fil This can significantly increase the tim taken to create the stream, particularly with a large file.
BASS_STREAM_AUTOFREE	Automatically free the stream when playback ends.
BASS_STREAM_DECODE	Decode the sample data, without playi

	it. Use BASS_ChannelGetData to retr decoded sample data. The BASS_SAMPLE_3D, BASS_STREAM_AUTOFREE and SPEAKER flags cannot be used togetl with this flag. The BASS_SAMPLE_SOFTWARE and BASS_SAMPLE_FX flags are also ignored.
BASS_SPEAKER_*xxx*	Speaker assignment flags. These flags have no effect when the stream is mor than stereo.
BASS_ASYNCFILE	Read the file asynchronously. When enabled, the file is read and buffered i parallel with the decoding, to reduce t chances of the decoder being affected I/O delays. This can be particularly us with slow storage media and/or low latency output. The size of the file buf is determined by the BASS_CONFIG_ASYNCFILE_BUF config option. This flag is ignored wh streaming from memory (*mem* = *TRU*
BASS_UNICODE	*file* is in UTF-16 form. Otherwise it is ANSI on Windows or Windows CE, a UTF-8 on other platforms.

Return value
If successful, the new stream's handle is returned, else 0 is returned. Use BASS_ErrorGetCode to get the error code.

Error codes

BASS_ERROR_INIT	BASS_Init has not been successfully called.
BASS_ERROR_NOTAVAIL	Only decoding channels (BASS_STREAM_DECODE) are allowed when using the "no sound" device. The BASS_STREAM_AUTOFREE flag is also unavailable to decoding channels.
BASS_ERROR_ILLPARAM	The *length* must be specified when streaming from memory.
BASS_ERROR_FILEOPEN	The file could not be opened.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_FORMAT	The sample format is not supported by the device/drivers. If the stream is more than stereo or the BASS_SAMPLE_FLOAT flag is used, it could be that they are not supported.
BASS_ERROR_SPEAKER	The specified SPEAKER flags are invalid. The device/drivers do not support them, they are attempting to assign a stereo stream to a mono speaker or 3D functionality is enabled.
BASS_ERROR_MEM	There is insuffcent memory.
BASS_ERROR_NO3D	Could not initialize 3D support.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks
Use BASS_ChannelGetInfo to retrieve information on the format of the stream. Opus always has a sample rate of 48000 Hz, but the source material may have had a different sample rate, which is available via the BASS_ATTRIB_OPUS_ORIGFREQ attribute. The playback length of the stream can be retrieved using BASS_ChannelGetLength.

The Opus file format is Ogg-based, so the standard BASS_TAG_OGG and BASS_TAG_VENDOR tag types apply to Opus too, via BASS_ChannelGetTags.

Chained Opus files containing multiple logical bitstreams are supported, but seeking within them is only fully supported if the BASS_STREAM_PRESCAN flag is used (or the BASS_CONFIG_OGG_PRESCAN option is enabled) to have them pre-scanned. Without pre-scanning, seeking will only be possible back to the start. The BASS_POS_OGG "mode" can be used with BASS_ChannelGetLength to get the number of bitstreams and with BASS_ChannelSetPosition to seek to a particular one. A BASS_SYNC_OGG_CHANGE sync can be set via BASS_ChannelSetSync to be informed of when a new bitstream begins during decoding/playback.

To stream a file from the internet, use BASS_OPUS_StreamCreateURL. To stream from other locations, see BASS_OPUS_StreamCreateFileUser.

Platform-specific

Away from Windows, all mixing is done in software (by BASS), so the BASS_SAMPLE_SOFTWARE flag is unnecessary. The BASS_SAMPLE_FX flag is also ignored.

Example

Create a stream of an Opus file.

```
HSTREAM stream=BASS_OPUS_StreamCreateFile(FALSE, "afile.opus", 0, 0
```

See also
BASS_OPUS_StreamCreateFileUser, BASS_OPUS_StreamCreateURL

BASS_ChannelGetInfo, BASS_ChannelGetLength, BASS_ChannelGetTags, BASS_ChannelPlay, BASS_ChannelSetAttribute, BASS_ChannelSetDSP, BASS_ChannelSetFX, BASS_ChannelSetLink, BASS_StreamFree, BASS_StreamGetFilePosition

BASS_OPUS_StreamCreateFileUser

Creates a sample stream from a Opus file via user callback functions.

```
HSTREAM BASS_OPUS_StreamCreateFileUser(
    DWORD system,
    DWORD flags,
    BASS_FILEPROCS *procs,
    void *user
);
```

Parameters

system	File system to use, one of the following.	
	STREAMFILE_NOBUFFER	Unbuffered.
	STREAMFILE_BUFFER	Buffered.
	STREAMFILE_BUFFERPUSH	Buffered, with the data pushed to BA via BASS_StreamPutFileData.
flags	A combination of these flags.	
	BASS_SAMPLE_FLOAT	Use 32-bit floating-point sample data Floating-point channels for more info this flag is not specified, then the stre is 16-bit.
	BASS_SAMPLE_SOFTWARE	Force the stream to not use hardware mixing.
	BASS_SAMPLE_3D	Enable 3D functionality. This require that the BASS_DEVICE_3D flag wa specified when calling BASS_Init, ar the stream must be mono. The SPEA flags can not be used together with th flag.
	BASS_SAMPLE_LOOP	Loop the file. This flag can be toggle any time using BASS_ChannelFlags.
	BASS_SAMPLE_FX	Enable the old implementation of Dir 8 effects. See the DX8 effect implementations section for details. U BASS_ChannelSetFX to add effects t stream.
	BASS_STREAM_PRESCAN	Pre-scan the file for seek points and accurate length reading in chained Op files (has no effect on normal Opus fi This can significantly increase the tin taken to create the stream, particularl with a large file. This flag only applie when using the STREAMFILE_NOBUFFER system

BASS_STREAM_RESTRATE	Restrict the "download" rate of the fil the rate required to sustain playback. this flag is not used, then the file will downloaded as quickly as possible. T flag only has effect when using the STREAMFILE_BUFFER system.
BASS_STREAM_BLOCK	Download and play the file in smaller chunks. Uses a lot less memory than otherwise, but it is not possible to see loop the stream; once it has ended, th must be opened again to play it again This flag will automatically be applie when the file length is unknown. This also has the effect of restricting the download rate. This flag has no effec when using the STREAMFILE_NOBUFFER system
BASS_STREAM_AUTOFREE	Automatically free the stream when playback ends.
BASS_STREAM_DECODE	Decode the sample data, without play it. Use BASS_ChannelGetData to ret decoded sample data. The BASS_SAMPLE_3D, BASS_STREAM_AUTOFREE and SPEAKER flags can not be used toge with this flag. The BASS_SAMPLE_SOFTWARE and BASS_SAMPLE_FX flags are also ignored.
BASS_SPEAKER_*xxx*	Speaker assignment flags. These flag have no effect when the stream is mo than stereo.
BASS_ASYNCFILE	Read the file asynchronously. When enabled, the file is read and buffered parallel with the decoding, to reduce chances of the decoder being affectec

	I/O delays. This can be particularly u with slow storage media and/or low latency output. The size of the file bu is determined by the BASS_CONFIG_ASYNCFILE_BU config option. This flag only applies using the STREAMFILE_NOBUFFE system.
procs	The user defined file functions.
user	User instance data to pass to the callback functions.

Return value
If successful, the new stream's handle is returned, else 0 is returned. Use BASS_ErrorGetCode to get the error code.

Error codes

BASS_ERROR_INIT	BASS_Init has not been successfully called.
BASS_ERROR_NOTAVAIL	Only decoding channels (BASS_STREAM_DECODE) are allowed when using the "no sound" device. The BASS_STREAM_AUTOFREE flag is also unavailable to decoding channels.
BASS_ERROR_ILLPARAM	*system* is not valid.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_FORMAT	The sample format is not supported by the device/drivers. If the stream is more than stereo or the BASS_SAMPLE_FLOAT flag is used, it could be that they are not supported.
BASS_ERROR_SPEAKER	The specified SPEAKER flags are invalid. The device/drivers do not support them, they are attempting to assign a stereo stream to a mono speaker or 3D functionality is enabled.
BASS_ERROR_MEM	There is insufficent memory.
BASS_ERROR_NO3D	Could not initialize 3D support.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks
When using a buffered file system, the playback length will not be available until the entire file has been "downloaded" via the file functions.

Platform-specific
Away from Windows, all mixing is done in software (by BASS), so the BASS_SAMPLE_SOFTWARE flag is unnecessary. The BASS_SAMPLE_FX flag is also ignored.

See also
BASS_OPUS_StreamCreateFile, BASS_OPUS_StreamCreateURL

BASS_ChannelGetInfo, BASS_ChannelGetLength, BASS_ChannelGetTags, BASS_ChannelPlay, BASS_ChannelSetAttribute, BASS_ChannelSetDSP, BASS_ChannelSetFX, BASS_ChannelSetLink, BASS_StreamFree, BASS_FILEPROCS structure, BASS_CONFIG_NET_BUFFER

BASS_OPUS_StreamCreateURL

Creates a sample stream from an Opus file on the internet, optionally receiving the downloaded data in a callback.

```
HSTREAM BASS_OPUS_StreamCreateURL(
    char *url,
    DWORD offset,
    DWORD flags,
    DOWNLOADPROC *proc,
    void *user
);
```

Parameters

url	URL of the file to stream. Should begin with "http://" or "https://" or "ftp://".
offset	File position to start streaming from. This is ignored by some servers, specifically when the file length is unknown.
flags	A combination of these flags.

BASS_SAMPLE_FLOAT	Use 32-bit floating-point sample data. See Floating-point channels for more info. If this flag is not specified, then the stream is 16-bit.
BASS_SAMPLE_SOFTWARE	Force the stream to not use hardware mixing.
BASS_SAMPLE_3D	Enable 3D functionality. This requires that the BASS_DEVICE_3D flag was specified when calling BASS_Init, and the stream must be mono. The SPEAKER flags can not be used together with this flag.
BASS_SAMPLE_LOOP	Loop the file. This flag can be toggled at any time using BASS_ChannelFlags. This flag is ignored when streaming in blocks (BASS_STREAM_BLOCK).
BASS_SAMPLE_FX	Enable the old implementation of DirectX 8 effects. See the DX8 effect implementations section for details. Use BASS_ChannelSetFX to add effects to the stream.
BASS_STREAM_RESTRATE	Restrict the download rate of the file to the rate required to sustain playback. If this flag is not used, then the file will be downloaded as quickly as the user's internet connection allows.
BASS_STREAM_BLOCK	Download and play the file in

	smaller chunks. Uses a lot less memory than otherwise, but it's not possible to seek or loop the stream; once it's ended, the file must be opened again to play it again. This flag will automatically be applied when the file length is unknown, for example with Shout/Icecast streams. This flag also has the effect of resticting the download rate.
BASS_STREAM_STATUS	Pass status info (HTTP/ICY tags) from the server to the DOWNLOADPROC callback during connection. This can be useful to determine the reason for a failure.
BASS_STREAM_AUTOFREE	Automatically free the stream when playback ends.
BASS_STREAM_DECODE	Decode the sample data, without playing it. Use BASS_ChannelGetData to retrieve decoded sample data. The BASS_SAMPLE_3D, BASS_STREAM_AUTOFREE and SPEAKER flags can not be used together with this flag. The BASS_SAMPLE_SOFTWARE and BASS_SAMPLE_FX flags are also ignored.
BASS_SPEAKER_*xxx*	Speaker assignment flags. These flags have no effect when the stream is more than stereo.
BASS_UNICODE	*url* is in UTF-16 form. Otherwise it is ANSI on Windows or Windows CE, and UTF-8 on other platforms.

proc Callback function to receive the file as it is downloaded... NULL = no

	callback.
user	User instance data to pass to the callback function.

Return value
If successful, the new stream's handle is returned, else 0 is returned. Use BASS_ErrorGetCode to get the error code.

Error codes

BASS_ERROR_INIT	BASS_Init has not been successfully called.
BASS_ERROR_NOTAVAIL	Only decoding channels (BASS_STREAM_DECODE) are allowed when using the "no sound" device. The BASS_STREAM_AUTOFREE flag is also unavailable to decoding channels.
BASS_ERROR_NONET	No internet connection could be opened.
BASS_ERROR_ILLPARAM	*url* is not a valid URL.
BASS_ERROR_SSL	SSL/HTTPS support is not available.
BASS_ERROR_TIMEOUT	The server did not respond to the request within the timeout period, as set with the BASS_CONFIG_NET_TIMEOUT config option.
BASS_ERROR_FILEOPEN	The file could not be opened.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_FORMAT	The sample format is not supported by the device/drivers. If the stream is more than stereo or the BASS_SAMPLE_FLOAT flag is used, it could be that they are not supported (ie. no WDM drivers).
BASS_ERROR_SPEAKER	The specified SPEAKER flags are invalid. The device/drivers do not support them, they are attempting to assign a stereo stream to a mono speaker or 3D functionality is enabled.
BASS_ERROR_MEM	There is insufficent memory.
BASS_ERROR_NO3D	Could not initialize 3D support.
BASS_ERROR_UNKNOWN	Some other mystery problem!

Remarks
Use BASS_ChannelGetInfo to retrieve information on the format of the stream. Opus always has a sample rate of 48000 Hz, but the source material may have had a different sample rate, which is available via the BASS_ATTRIB_OPUS_ORIGFREQ attribute. The playback length is not available until the entire file has been downloaded, at which point it can be retrieved using BASS_ChannelGetLength.

The Opus file format is Ogg-based, so the standard BASS_TAG_OGG and BASS_TAG_VENDOR tag types apply to Opus too, via BASS_ChannelGetTags. The BASS_SYNC_OGG_CHANGE sync is also supported, via BASS_ChannelSetSync.

When playing the stream, BASS will stall the playback if there is insufficient data to continue playing. Playback will automatically be resumed when sufficient data has been downloaded. BASS_ChannelIsActive can be used to check if the playback is stalled, and the progress of the file download can be checked with BASS_StreamGetFilePosition.

When streaming in blocks (BASS_STREAM_BLOCK flag), be careful not to stop/pause the stream for too long, otherwise the connection may timeout due to there being no activity and the stream will end prematurely.

When using an *offset*, the file length returned by BASS_StreamGetFilePosition can be used to check that it was successful by comparing it with the original file length. Another way to check is to inspect the HTTP headers retrieved with BASS_ChannelGetTags.

Platform-specific
Away from Windows, all mixing is done in software (by BASS), so the BASS_SAMPLE_SOFTWARE flag is unnecessary. The BASS_SAMPLE_FX flag is also ignored.

See also
BASS_OPUS_StreamCreateFile, BASS_OPUS_StreamCreateFileUser

BASS_ChannelGetInfo, BASS_ChannelGetLength, BASS_ChannelGetTags, BASS_ChannelPlay, BASS_ChannelSetAttribute, BASS_ChannelSetDSP, BASS_ChannelSetFX, BASS_ChannelSetLink, BASS_StreamFree, DOWNLOADPROC callback, BASS_CONFIG_NET_AGENT, BASS_CONFIG_NET_BUFFER, BASS_CONFIG_NET_PREBUF, BASS_CONFIG_NET_PROXY, BASS_CONFIG_NET_TIMEOUT

Plugin system

As well as providing dedicated stream creation functions, BASSOPUS supports the BASS plugin system, adding Opus file support to the standard BASS stream and sample creation functions: BASS_StreamCreateFile, BASS_StreamCreateFileUser, and BASS_SampleLoad. This is enabled using the BASS_PluginLoad function.

BASS_ATTRIB_OPUS_GAIN attribute

The output gain of an Opus stream.

```
BASS_ChannelGetAttribute(
    HSTREAM handle,
    BASS_ATTRIB_OPUS_GAIN,
    float *gain
);
```

Parameters

handle	The Opus stream handle.
gain	The gain in dB.

Remarks

Opus files have an "output gain" header field, which is applied by BASSOPUS to the decoded sample data. This attribute can be used to retrieve and override that gain value. When there are multiple logical bitstreams, each bitstream has its own output gain value, and this attribute will be reset to the new bitstream's header value upon a bitstream switch. A BASS_SYNC_OGG_CHANGE sync can be set via BASS_ChannelSetSync to be informed of when a new bitstream begins during decoding/playback.

See also
BASS_ChannelGetAttribute, BASS_ChannelSetAttribute

BASS_ATTRIB_OPUS_ORIGFREQ attribute

The sample rate of an Opus stream's source material.

```
BASS_ChannelGetAttribute(
    HSTREAM handle,
    BASS_ATTRIB_OPUS_ORIGFREQ,
    float *freq
);
```

Parameters

handle	The Opus stream handle.
freq	The sample rate.

Remarks

Opus streams always have a sample rate of 48000 Hz, and an Opus encoder will resample the source material to that if necessary. This attribute presents the original sample rate, which may be stored in the Opus file header. This attribute is read-only, so cannot be modified via BASS_ChannelSetAttribute.

See also
BASS_ChannelGetAttribute