

# BASS\_WV\_StreamCreateFile

---

Creates a sample stream from a WavPack file.

```
HSTREAM BASS_WV_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. <a href="#">Floating-point channels</a> for info.
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 <a href="#">BASS_Init</a> , 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 any time using <a href="#">BASS_ChannelFlags</a> .
BASS_SAMPLE_FX	Enable the old implementation of Direct8 effects. See the <a href="#">DX8 effect implementations</a> section for details. Use <a href="#">BASS_ChannelSetFX</a> to add effects to stream.
BASS_STREAM_AUTOFREE	Automatically free the stream when playback ends.
BASS_STREAM_DECODE	Decode the sample data, without playing it. Use <a href="#">BASS_ChannelGetData</a> to retrieve decoded sample data. The BASS_SAMPLE_3D, BASS_STREAM_AUTOFREE and SPEAKER flags cannot be used together with this flag. The BASS_SAMPLE_SOFTWARE and BASS_SAMPLE_FX flags are also

BASS_SPEAKER_xxx	ignored. <a href="#">Speaker assignment flags</a> . These flags have no effect when the stream is more than stereo.
BASS_ASYNCFILE	Read the file asynchronously. When enabled, the file is read and buffered in parallel with the decoding, to reduce the chances of the decoder being affected by I/O delays. This can be particularly useful with slow storage media and/or low latency output. The size of the file buffer is determined by the <a href="#">BASS_CONFIG_ASYNCFILE_BUFFER</a> config option. This flag is ignored when streaming from memory ( <i>mem = TRUE</i> ). If the <i>file</i> is in UTF-16 form. Otherwise it is ANSI on Windows or Windows CE, a UTF-8 on other platforms.
BASS_UNICODE	

**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	<a href="#">BASS_Init</a> 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 <i>length</i> 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_CODEEC	The BASSDSD add-on needs to be loaded to handle DSD data.
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 insufficient memory.
BASS_ERROR_NO3D	Could not initialize 3D support.
BASS_ERROR_UNKNOWN	Some other mystery problem!

## Remarks

All WavPack sample resolutions from 8 to 32-bit are supported, but the output will be restricted to 16-bit unless the `BASS_SAMPLE_FLOAT` flag is used. The file's original resolution is available via [BASS\\_ChannelGetInfo](#). WavPack hybrid files are supported, and a "correction" (WVC) file will automatically be used if present. WavPack DSD files are also supported if the BASSDSD add-on (v2.4.1 or above) is loaded.

Use [BASS\\_ChannelGetInfo](#) to retrieve information on the format (sample rate, resolution, channels) of the stream. The playback length of the stream can be retrieved using [BASS\\_ChannelGetLength](#). A DSD file can be identified by the [BASS\\_ATTRIB\\_DSD\\_RATE](#) attribute being available.

WavPack files can have ID3 or APEv2 tags, which can be retrieved by using `BASS_TAG_ID3` or `BASS_TAG_APE` with [BASS\\_ChannelGetTags](#).

To stream a file from the internet, use [BASS\\_WV\\_StreamCreateURL](#). To stream from other locations, see [BASS\\_WV\\_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 a WavPack file.

```
HSTREAM stream=BASS_WV_StreamCreateFile(FALSE, "afile.wv", 0, 0, 0)
```

**See also**

[BASS\\_WV\\_StreamCreateFileUser](#), [BASS\\_WV\\_StreamCreateURL](#)

[BASS\\_ChannelGetInfo](#), [BASS\\_ChannelGetLength](#), [BASS\\_ChannelGetTags](#),  
[BASS\\_ChannelPlay](#), [BASS\\_ChannelSetAttribute](#), [BASS\\_ChannelSetDSP](#),  
[BASS\\_ChannelSetFX](#), [BASS\\_ChannelSetLink](#), [BASS\\_StreamFree](#),  
[BASS\\_StreamGetFilePosition](#)

# BASS\_WV\_StreamCreateFileUser

---

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

```
HSTREAM BASS_WV_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 <a href="#">BASS_StreamPutFileData</a> .
flags	A combination of these flags.
BASS_SAMPLE_FLOAT	Use 32-bit floating-point sample data <a href="#">Floating-point channels</a> for more info.
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 <a href="#">BASS_Init</a> , 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 any time using <a href="#">BASS_ChannelFlags</a> .
BASS_SAMPLE_FX	Enable the old implementation of Dir 8 effects. See the <a href="#">DX8 effect implementations</a> section for details. Use <a href="#">BASS_ChannelSetFX</a> 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 possible. This 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 seek loop the stream; once it has ended, the

must be opened again to play it again. This flag will automatically be applied when the file length is unknown. This also has the effect of restricting the download rate. This flag has no effect when using the `STREAMFILE_NOBUFFER` system.

`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_ASYNCFILE` Read the file asynchronously. When enabled, the file is read and buffered in parallel with the decoding, to reduce the chances of the decoder being affected by I/O delays. This can be particularly useful with slow storage media and/or low latency output. The size of the file buffer is determined by the [BASS\\_CONFIG\\_ASYNCFILE\\_BUFFER](#) config option. This flag only applies when using the `STREAMFILE_NOBUFFER` 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	<a href="#">BASS_Init</a> 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	<i>system</i> is not valid.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_CODEEC	The BASSDSD add-on needs to be loaded to handle DSD data.
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 insufficient memory.
BASS_ERROR_NO3D	Could not initialize 3D support.
BASS_ERROR_UNKNOWN	Some other mystery problem!

**Remarks**

WavPack hybrid files will be decoded in their lossy form, as any "correction" file is inaccessible.

**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\\_WV\\_StreamCreateFile](#), [BASS\\_WV\\_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\_WV\_StreamCreateURL

---

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

```
HSTREAM BASS_WV_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 <a href="#">Floating-point channels</a> for more info.
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 <a href="#">BASS_Init</a> , 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 <a href="#">BASS_ChannelFlags</a> . This flag is ignored when streaming in blocks (BASS_STREAM_BLOCK).
BASS_SAMPLE_FX	Enable the old implementation of DirectX 8 effects. See the <a href="#">DX8 effect implementations</a> section for details. Use <a href="#">BASS_ChannelSetFX</a> 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 restricting the download rate.

BASS_STREAM_STATUS	Pass status info (HTTP/ICY tags) from the server to the <a href="#">DOWNLOADPROC</a> 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 <a href="#">BASS_ChannelGetData</a> 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	<a href="#">Speaker assignment flags</a> . These flags have no effect when the stream is more than stereo.
BASS_UNICODE	<i>url</i> 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	<a href="#">BASS_Init</a> 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	<i>url</i> 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 <a href="#">BASS_CONFIG_NET_TIMEOUT config option</a> .
BASS_ERROR_FILEOPEN	The file could not be opened.
BASS_ERROR_FILEFORM	The file's format is not recognised/supported.
BASS_ERROR_CODEEC	The BASSDSD add-on needs to be loaded to handle DSD data.
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 insufficient 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 (sample rate, resolution, channels) of the stream. The playback length of the stream can be retrieved using [BASS\\_ChannelGetLength](#). If the file's header does not contain length information, then the length will not be available until the entire file has been streamed.

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\\_ChannelsActive](#) 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\\_WV\\_StreamCreateFile](#), [BASS\\_WV\\_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, BASSWV supports the BASS plugin system, adding WavPack 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.